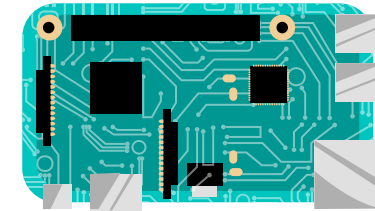


# Raspberry PI 3 y pcDUINO

Tutorial de instalación y configuración

## AUTORES

- María Fernanda Díaz Hernández
- José Luis González Montenegro
- Ruben Dario Hernández Beleño
- Jaime Durán García
- Natalia Sánchez Sánchez



# Raspberry Pi 3 y pcDUINO

Tutorial de instalación y configuración

María Fernanda Díaz Hernández  
José Luis González Montenegro  
Ruben Dario Hernández Beleño  
Jaime Durán García  
Natalia Sánchez Sánchez

## Universidad Piloto de Colombia

### Presidente

José María Cifuentes Páez

### Rectora

Ángela Gabriela Bernal Medina

### Director de Publicaciones y Comunicación Gráfica

Rodrigo Lobo-Guerrero Sarmiento

### Director de Investigaciones

Mauricio Hernández Tascón

### Coordinador General de Publicaciones

Diego Ramírez Bernal

### Decano del Programa de ingeniería mecatrónica

Jaime Durán García

### Diseño de portada y diagramación

Daniela Martínez Díaz

Laura C. Soto Velásquez

Departamento de Publicaciones y Comunicación Gráfica  
de la Universidad Piloto de Colombia

### Vectores portada tomados de:

<http://www.freepik.es/>

La obra literaria publicada expresa exclusivamente la opinión de sus respectivos autores, de manera que no representan el pensamiento de la Universidad Piloto de Colombia. Cada uno de los autores, suscribió con la Universidad una autorización o contrato de cesión de derechos y una carta de originalidad sobre su aporte, por tanto, los autores asumen la responsabilidad sobre el contenido de esta publicación

## Raspberry Pi 3 y pcDuino, tutorial de instalación y configuración

### ISBN

978-958-8957-74-6

Copyright ©

Primera edición - 2018

Bogotá, Colombia

### Autores

María Fernanda Díaz Hernández

José Luis González Montenegro

Ruben Dario Hernández Beleño

Jaime Durán García

Natalia Sánchez Sánchez

Díaz Hernández, María Fernanda

**Raspberry PI 3** : tutorial de instalación y configuración  
/ María Fernanda Díaz Hernández... ( y otros 4 )

**Bogotá** : Universidad Piloto de Colombia, 2018

**52 páginas** : ilustraciones

Incluye bibliografía ( páginas 50-51 )

ISBN: 9789588957746

**1. RASPBERRY PI ( COMPUTADOR ) - PROGRAMACION**

**2. MICROCOMPUTADORES - PROGRAMACION**

**3. SISTEMAS EMBEBIDOS DE COMPUTADOR**

CDD 005.3

**I.** González Montenegro, José Luis

**II.** Hernández Beleño, Ruben Dario

**III.** Duran García, Jaime

**IV.** Sánchez Sánchez, Natalia

# Contenido

|   |    |
|---|----|
| PRESENTACIÓN  | 15 |
| INTRODUCCIÓN  | 17 |
| LOS SISTEMAS EMBEBIDOS  | 18 |
| Raspberry Pi 3  | 20 |
| 1. Cómo hacer que funcione Raspberry Pi 3                                 | 21 |
| 1.1 Formatear el "disco duro"   | 22 |
| 1.2 Descargar el sistema operativo  | 27 |
| 1.3 Instalar el sistema operativo y el software aplicativo                | 32 |
| 1.4 Cambiar la configuración del teclado                                  | 44 |
| 2. Configuraciones adicionales  | 48 |
| 2.1 Instalar utilidades para manipular archivos .xls con Phyton           | 48 |
| 2.2 Configuración para subir archivos Dropbox desde Phyton                | 51 |
| 3. Aplicación: subir datos a la nube                                      | 54 |
| 3.1 El sensor DHT11   | 55 |
| 4. Aplicación: subir datos a la nube (detección de presencia de personas) | 61 |
| 4.1 Instalar librería para el puerto GPIO                                 | 64 |

|   |                |           |
|---|----------------|-----------|
|   | <b>pcDuino</b> | <b>68</b> |
| 1. Cómo hacer que funcione pcDuino  |                | <b>69</b> |
| 1.1 Cambiar configuración de teclado                                      |                | <b>74</b> |
| 2. Configuración adicional  |                | <b>75</b> |
| 2.1 Instalar utilidad de gestión de paquetes para Phyton                  |                | <b>75</b> |
| 2.2 Instalar utilidad con sistemas de control de versiones                |                | <b>76</b> |
| 2.3 Instalar utilidades para manipular archivos .xls con Phyton           |                | <b>76</b> |
| 2.4 Configurar para subir archivos a Dropbox desde Phyton                 |                | <b>77</b> |
| 3. Aplicación: subir datos a la nube (medición de temperatura /humedad)   |                | <b>80</b> |
| 3.1 Sensor DHT11  |                | <b>82</b> |
| 4. Aplicación: subir datos a la nube (detección de presencia de personas) |                | <b>86</b> |
| 4.1 Clonar librería para manipular puertos de entrada /salida             |                | <b>89</b> |



# Lista de Figuras



|   |           |
|---|-----------|
| ● <b>Figura 1.</b> Página de descarga de SD Card Formatter for Windows                  | <b>23</b> |
| ● <b>Figura 2.</b> Interfaz gráfica SD Formatter  | <b>24</b> |
| ● <b>Figura 3.</b> Confirmación del formato escogido para la SD Card                    | <b>25</b> |
| ● <b>Figura 4.</b> Confirmación del proceso de formato de la SD Card                    | <b>26</b> |
| ● <b>Figura 5.</b> Confirmación de la finalización del proceso de formato de la SD Card | <b>26</b> |
| ● <b>Figura 6.</b> Página de descarga del sistema operativo Raspberry Pi                | <b>29</b> |
| ● <b>Figura 7.</b> Interfaz del explorador de Windows, carpeta de descargas             | <b>30</b> |
| ● <b>Figura 8.</b> Ruta para descomprimir el sistema operativo Raspberry Pi             | <b>30</b> |
| ● <b>Figura 9.</b> Carpeta de instalación para Raspberry Pi                             | <b>31</b> |
| ● <b>Figura 10.</b> Instalación de la SD Card en el sistema embebido Raspberry Pi       | <b>32</b> |
| ● <b>Figura 11.</b> Interfaz de instalación de NOOBS para Raspberry Pi                  | <b>33</b> |
| ● <b>Figura 12.</b> Procedimiento de instalación sistema operativo Raspbian             | <b>34</b> |
| ● <b>Figura 13.</b> Finalización de instalación de Raspbian en la SD Card               | <b>35</b> |

|   |    |
|---|----|
| • <b>Figura 14.</b> Descripción del escritorio de trabajo de Raspbian               | 36 |
| • <b>Figura 15.</b> Panel de configuración de periféricos de Raspbian               | 45 |
| • <b>Figura 16.</b> Terminal del sistema operativo Raspbian                         | 47 |
| • <b>Figura 17.</b> Instalación de paquetes desde el terminal del sistema operativo | 49 |
| • <b>Figura 18.</b> Configuración para subir archivos a Dropbox en modo programador | 51 |
| • <b>Figura 19.</b> Instalación del paquete de Dropbox en Raspbian                  | 52 |
| • <b>Figura 20.</b> Sensor de temperatura y humedad DHT11                           | 55 |
| • <b>Figura 21.</b> Esquema de conexión de Raspberry Pi con el sensor DHT11         | 57 |
| • <b>Figura 22.</b> Numeración de los pines de Raspberry Pi                         | 58 |
| • <b>Figura 23.</b> Sensor de presencia HC-SR501                                    | 62 |
| • <b>Figura 24.</b> Esquema de conexión del sensor de presencia con Raspberry Pi    | 63 |
| • <b>Figura 25.</b> Esquema de pines GPIO para conexión con Raspsberry Pi           | 64 |
| • <b>Figura 26.</b> Diagrama de configuración de pcDuino                            | 71 |
| • <b>Figura 27.</b> Interfaz gráfica del sistema operativo de pcDuino               | 72 |
| • <b>Figura 28.</b> Terminal de configuración del sistema operativo de pcDuino      | 74 |
| • <b>Figura 29.</b> Instalación del paquete de Dropbox en pcDuino                   | 78 |
| • <b>Figura 30.</b> Componentes electrónicos para el desarrollo de la aplicación    | 81 |
| • <b>Figura 31.</b> Sensor de temperatura y humedad DHT11                           | 82 |

|   |    |
|---|----|
| • <b>Figura 32.</b> Montaje entre el pcDuino y el sensor de DTH11                   | 83 |
| • <b>Figura 33.</b> Dispositivos de conexión del sensor de presencia y pcDuino      | 87 |
| • <b>Figura 34.</b> Sensor de presencia HC-SR501                                    | 88 |
| • <b>Figura 35.</b> Montaje sensor de presencia y pcDuino                           | 89 |
| • <b>Figura 36.</b> Montaje de conexión entre pcDuino y el computador de escritorio | 90 |



# PRESENTACIÓN

En buena hora llega a los jóvenes colombianos, estudiantes de ingeniería y a la comunidad académica un mapa didáctico, como cartilla interactiva, producto de la evaluación técnica y pedagógica de reconocidos profesionales de la Universidad Piloto de Colombia, quienes han identificado una de las motivaciones más importantes del mundo de la ingeniería “el aprender haciendo”.

El propósito de este diseño metodológico, en forma de cartilla de estrategias didácticas, es la de orientar a los estudiantes de ingeniería, en especial a los mecatrónicos y los electrónicos, interesados en procesos de automatización, sensórica y robótica, para que su aprendizaje se afiance en cada uno de los pasos necesarios para la instalación y configuración de sistemas embebidos: Raspberry Pi 3 y pcDuino; el trabajo consignado está dirigido a los estudiantes y a todos aquellos docentes mediadores que orientan cátedras en el área de tecnología.

Teniendo en cuenta que desde hace un par de años estamos viviendo en la cuarta revolución industrial, los lectores de esta cartilla tendrán la oportunidad de surfear, pero también bucear en los contenidos planteados, desde donde se tendrá la aplicación de los estándares que permiten propiciar el desarrollo de diversas actividades que ayuden a incentivar a



los estudiantes de ingeniería por el aprendizaje guiado e interactivo que propicie pasar de modelos de automatización industrial a una automatización digital.

Los lectores encontrarán que los conocimientos aquí compartidos son esenciales en el desempeño profesional y sin ellos será imposible competir en actividades laborales. Es fácil identificar que los autores, para realizar éste trabajo, han partido de la observación, lo que da mucho sentido a la comprensión y apropiación de los contenidos, pues evidencia que las metodologías dadas por los otros textos similares no han sido suficiente, razón por la cual no se han logrado los objetivos de enseñanza - aprendizaje necesarios para dar independencia de pensamiento ingenieril.

De esta manera, a través de este manual, se desea desde la Universidad Piloto de Colombia, contribuir a superar tal situación, dejando esta cartilla conformada por estrategias orientadas al desarrollo de las competencias básicas, abordando los estándares de un aprendizaje innovador bajo una manera más amena y didáctica. Los fundamentos teóricos están basados en el constructivismo, así como los trabajos de Vygotsky que hacen referente al diagnóstico estructural del conocimiento y,

desde la perspectiva ingenieril, con Ausubel en el aprendizaje significativo.

El lector se adentrará en un conocimiento dinámico el cual pretende que los lectores se conviertan en estudiantes activos, que usan esta poderosa herramienta en forma de cartilla, para impactar las diferentes materias que conforman los planes de estudio de las carreras de ingeniería, sin dejar de lado los aportes para los sectores industrial, comercial y doméstico presentes en las diferentes situaciones.

Igualmente, la cartilla motiva a su lectura, por cuanto las nuevas tecnologías de la información y las comunicaciones se han involucrado en todos los aspectos de la vida de las personas y, los temas tratados se encuentran inmersas (embebidas) desde muchos dispositivos usados por muchos sectores, por ejemplo para el sector de salud, como en el caso de los médicos, hasta una gran presencia en los electrodomésticos, equipos industriales y en los negocios. Por ser un manual no se detiene en los avances significativos presentes tanto en hardware como en software, pero si logra proveer elementos que permiten entender la esencia de programación libre y de alta prestación de servicios.

## INTRODUCCIÓN

Las nuevas tecnologías de la información y las comunicaciones se han involucrado en todos los aspectos de la vida de las personas. Se encuentran inmersas (embebidas) tanto en los dispositivos médicos como en los electrodomésticos; en los equipos industriales como en los negocios. Los avances son muy significativos tanto en *hardware* como en *software* y han logrado proveer a los usuarios de elementos de tamaño y costo muy reducidos, de programación libre y de alta prestación de servicios.

Esta cartilla, con la presentación de los pasos necesarios para la instalación y configuración de dos placas con sistemas embebidos Raspberry Pi 3 y pcDuino, está dirigida a estudiantes de ingeniería mecatrónica, robótica o programación interesados en la utilización de las nuevas tecnologías de la información y las comunicaciones para desarrollar propuestas relacionadas con casas inteligentes (domótica), obtención de datos mediante sensores, Internet en la nube e Internet de las cosas. Ha sido desarrollada dentro del ambiente académico y de investigación formativa del Programa de Ingeniería Mecatrónica de la Universidad Piloto de Colombia, cuyos componentes principales son el Taller Mecatrónico, los semilleros de investigación y la ayuda a poblaciones vulnerables desde los espacios de la Proyección Social.

A continuación, se presentan los pasos de instalación y configuración, seguidos de una aplicación de ejemplo de programación y funcionamiento para cada tarjeta, lo que dará a los lectores una información sencilla y completa, y les permitirá avanzar en el mundo de las aplicaciones para dar vía libre a su creatividad.

## LOS SISTEMAS EMBEBIDOS

Los sistemas embebidos hacen referencia a una combinación de *hardware* y *software* de computador diseñada para realizar operaciones de cómputo, normalmente en tiempo real, y cumplir con una función específica; sus recursos son limitados, pero son muy útiles en la construcción de productos dedicados. Esta combinación puede ser reemplazada en muchos casos por un circuito integrado que realice la misma tarea, pero una de las ventajas de los sistemas embebidos es su flexibilidad, ya que a la hora de realizar alguna modificación resulta mucho más sencillo modificar unas líneas de código en el *software* que reemplazar todo el circuito integrado (Salas, 2015, p. 38).

Existen básicamente dos tipos de sistemas embebidos: los que corresponden a sistemas autónomos y que funcionan por sí solos, capaces de realizar varias funciones a la vez; y los que son parte de sistemas mayores, los cuales cumplen una funcionalidad.

Los miniPC son siempre una buena opción para disfrutar de toda la potencia de un ordenador, pero en un tamaño compacto. Pueden

ser usados como proveedores de contenidos, conectarse a un televisor o como ordenadores personales. Entre los más populares se encuentra Raspberry Pi por su bajo costo y sus enormes posibilidades. Desde su lanzamiento en 2012 fue acogido por miles de entusiastas de los ordenadores de bajo costo que constituyen una comunidad que aporta valor al proyecto mediante innumerables aplicaciones que se pueden implementar con poco conocimiento. La placa base de la Raspberry Pi aloja un chip Broadcom BCM2835, procesador ARM de hasta 1 GHz, GPU VideoCore IV y 512 Mb de memoria RAM. Este embebido se desarrolló en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en el año 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque su comercialización empezó hasta el año 2012. El *software* que utiliza es *open source* y su sistema operativo oficial es el Raspbian,

sin embargo permite otros sistemas operativos, incluida una versión de Windows 10.

La Raspberry Pi no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa (Blog Historia de la Informática, 2013). No obstante, Raspberry Pi no es la única tarjeta que se encuentra en el mercado: pcDuino es una minicomputadora que puede correr Ubuntu o Android y, además, cuenta con herramientas compatibles con el Arduino. La pcDuino puede ser usada como una computadora y, como otros dispositivos de esta naturaleza, no tiene aceleración de gráficos por *hardware*, lo cual significa que podría haber afectación al reproducir videos o en el desempeño de gráficos 3D. Quizás la gran virtud de esta tarjeta/computadora es la relación estrecha que existe con los encabezados de Arduino, además que permite desarrollar proyectos y prototipos para distintas aplicaciones (Michelson, 2013).



# Raspberry Pi 3

## 1. Cómo hacer que funcione Raspberry Pi 3

### ● Presentación

Raspberry Pi es una placa de ordenador desarrollada en el Reino Unido por la fundación Raspberry Pi. La idea del proyecto se concibió en 2006 como una solución orientada a entornos educativos para estimular la enseñanza de ciencias de la computación en las escuelas.

Esta placa incluye procesador, memoria RAM, ranura para tarjetas SD, conectores USB, vídeo RCA, Conexión Jack de 3,5 pulgadas para audio, conector HDMI, entre otros; además, permite ejecutar Linux o sistemas RISC. Todo esto posibilita la ejecución de miles de aplicaciones con grandes beneficios y bajo costo.

### ● Materiales

- \* Microcomputador Raspberry Pi 3
- \* Memoria microSD ( $\geq 8$  Gb)
- \* Fuente de alimentación 5V, 1A con conector micro USB
- \* Teclado USB
- \* Monitor (conexión HDMI)
- \* Ratón USB (opcional)
- \* Cable Ethernet

## 1.1 Formatear el “disco duro”

- Para dar formato FAT32 a la tarjeta micro SD (el disco duro de nuestra Raspberry Pi), puede utilizarse el *software* SD Formatter (en Internet se encuentra una multitud de *software* que cumple la misma función). Puede descargarse de:

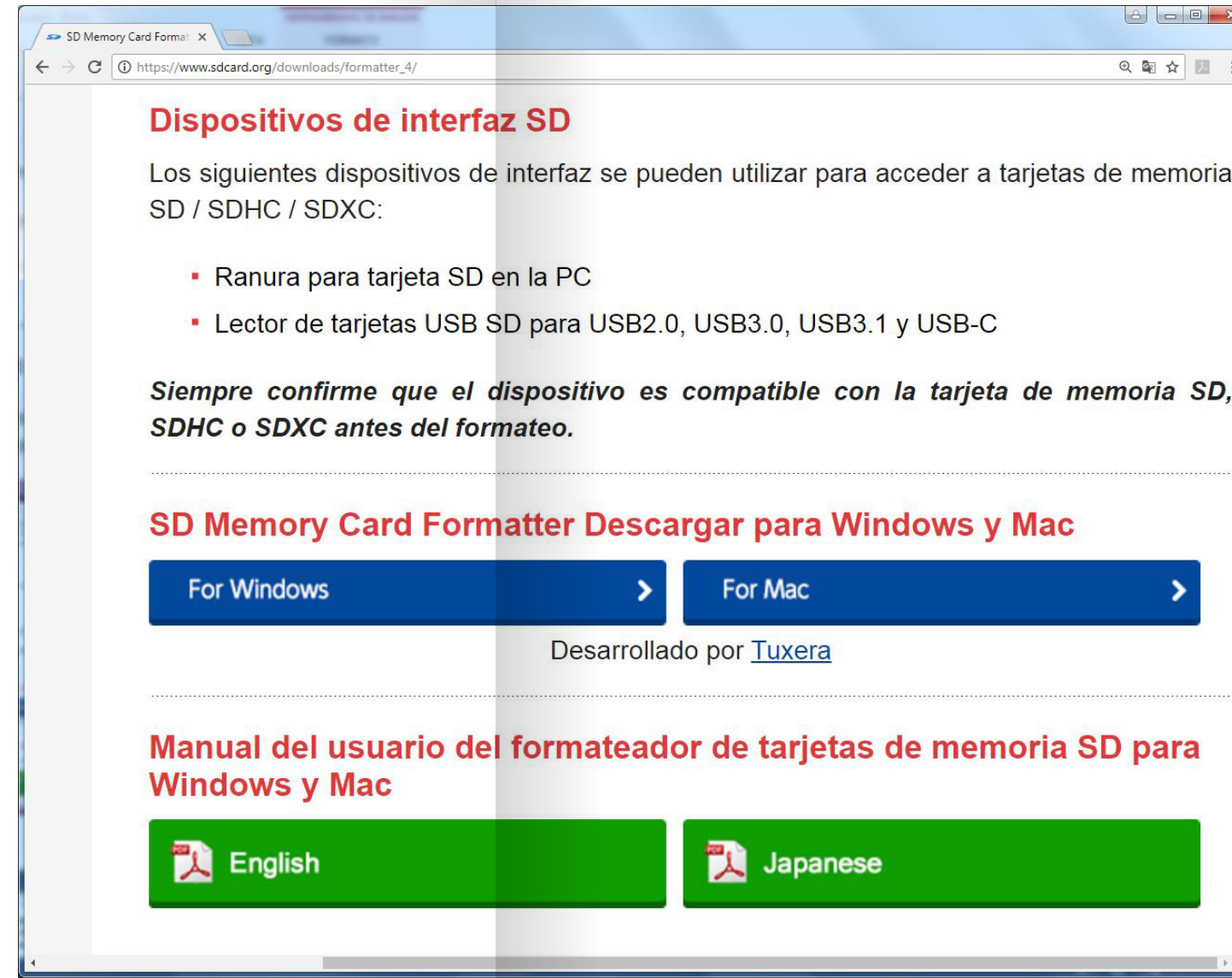
[https://www.sdcard.org/downloads/formatter\\_4/](https://www.sdcard.org/downloads/formatter_4/)



22

**Paso 1** En la parte inferior de la página, como se muestra en la figura 1, haga clic en el botón “Download SD Card Formatter for Windows >”

Cuando se descargue el programa completamente, instálelo con las opciones por defecto.



**Figura 1**

Página de descarga de SD Card  
Formatter for Windows

Fuente: Elaboración propia.

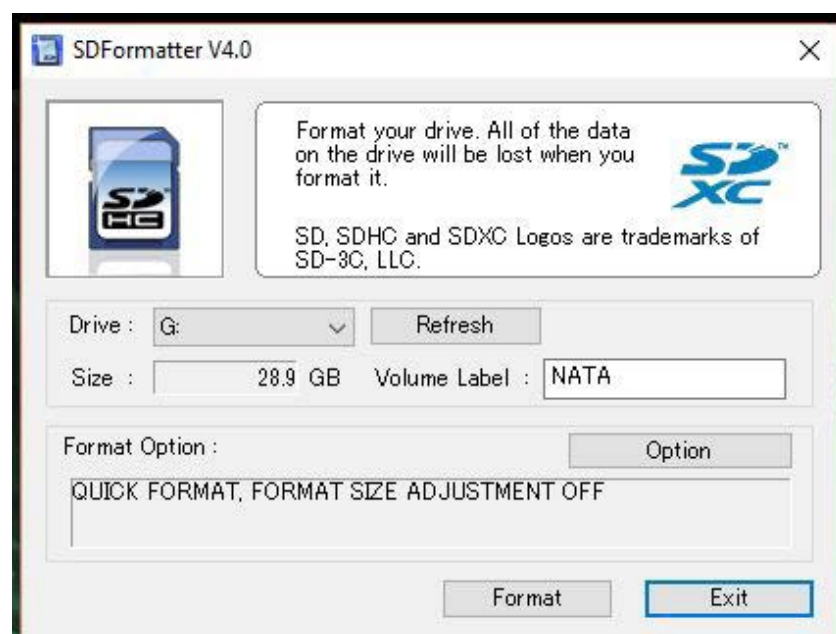
23

## Paso 2 Ejecutar el programa

**ADVERTENCIA:** este programa borra definitivamente los datos de la unidad seleccionada en el campo *Drive*.



Verifique que la letra del campo *Drive* corresponda con la letra asignada a la tarjeta micro SD y haga clic en el botón *Option*.



**Figura 2**

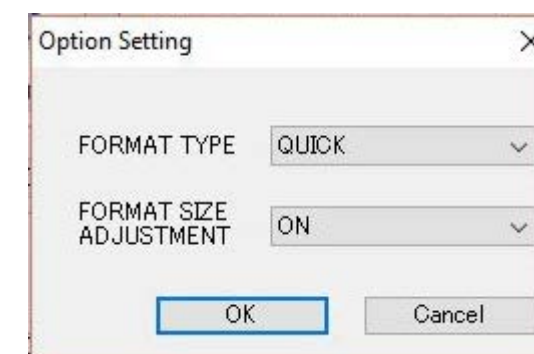
Interfaz gráfica SD Formatter  
Fuente: Elaboración propia.

## Paso 3 Seleccionar el tipo de formato

*Quick* (formato rápido)  
*Full* (formato completo)

En *Format Size Adjustment* seleccione *ON* y luego seleccione el botón *OK*.

**Nota:** si no aparece esta opción, haga clic en aceptar.

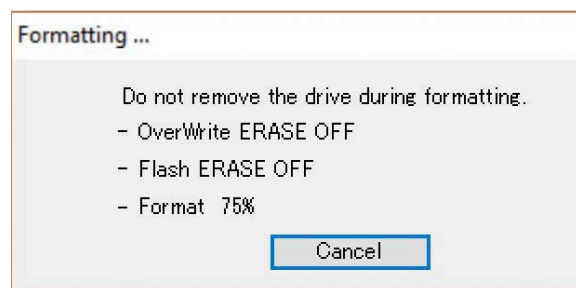


**Figura 3**

Confirmación del formato escogido para la SD Card  
Fuente: Elaboración propia.

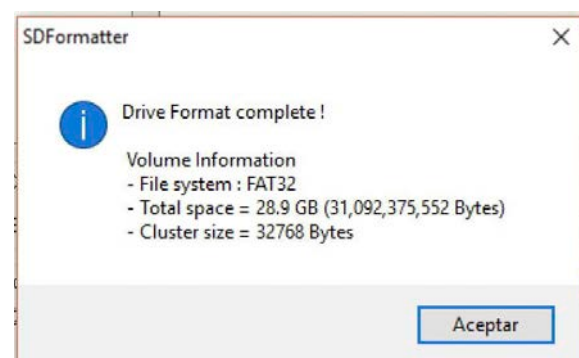


**Paso 4** En la ventana principal, seleccione el botón *Format* para iniciar el formateo de la memoria micro SD.



**Figura 4**  
Confirmación del proceso de  
formato de la SD Card  
Fuente: Elaboración propia.

**Paso 5** Cuando finalice el formateo, salga del programa, verifique el formato y extraiga la tarjeta micro SD.



**Figura 5**  
Confirmación de la finalización del  
proceso de formato de la SD Card  
Fuente: Elaboración propia.

## 1.2 Descargar el sistema operativo

**Nota aclaratoria:** Para la tarjeta Raspberry Pi existen varios Sistemas Operativos (SO) disponibles, pero es Raspbian el SO de diseño nativo para esta tarjeta. Es una distribución de Linux compilada especialmente para esta plataforma; para facilitar el proceso, existe un gestor de instalación llamado NOOBS. Puede descargarse de la página oficial de Raspberry:

<https://www.raspberrypi.org/downloads/noobs/>

Paso 1 En el campo NOOBS *Offline and network install*, seleccione el botón *Download ZIP*.

El archivo tiene un tamaño superior a 1Gb.

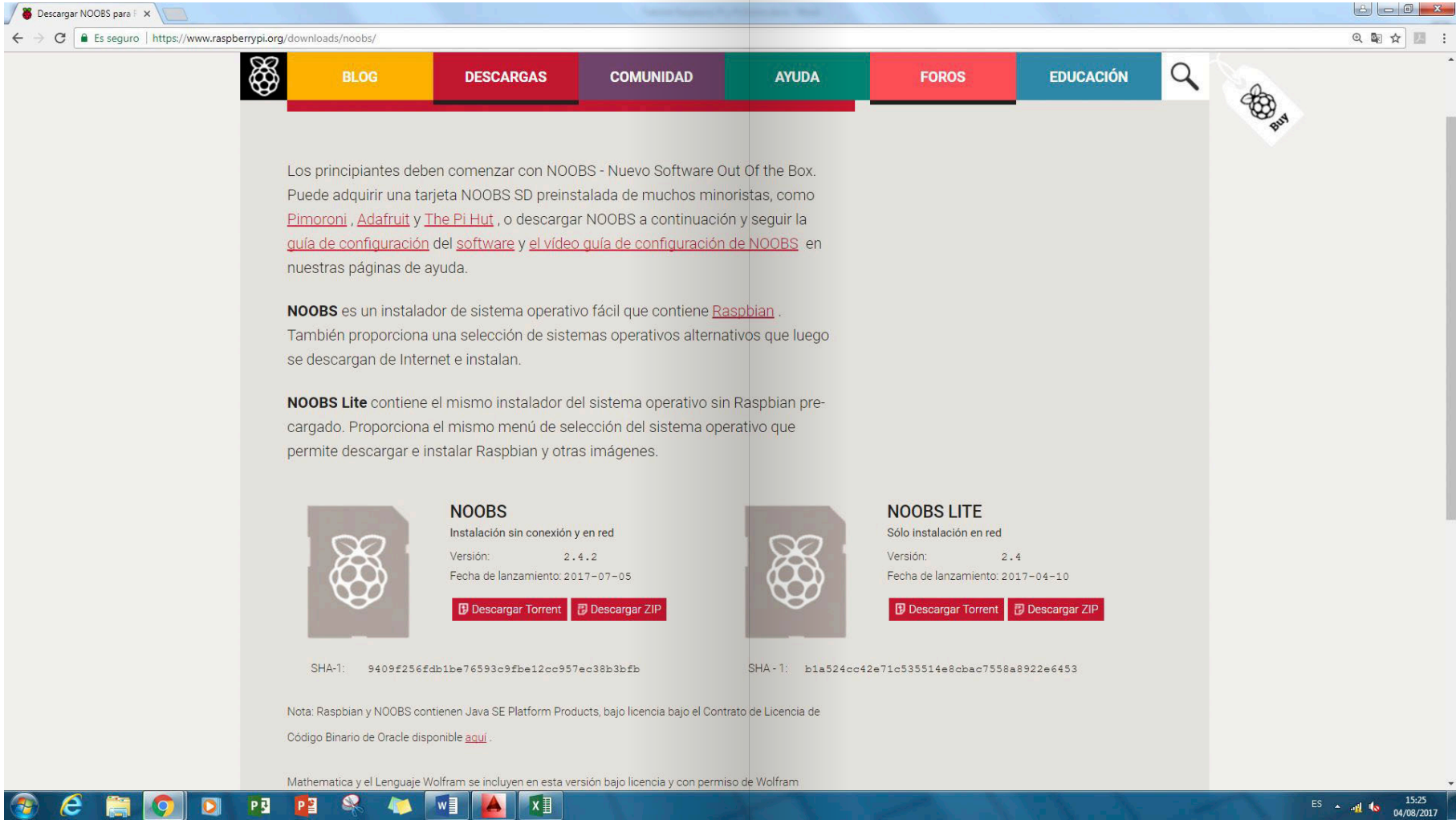
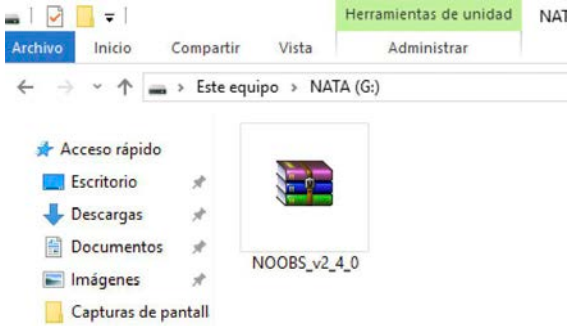


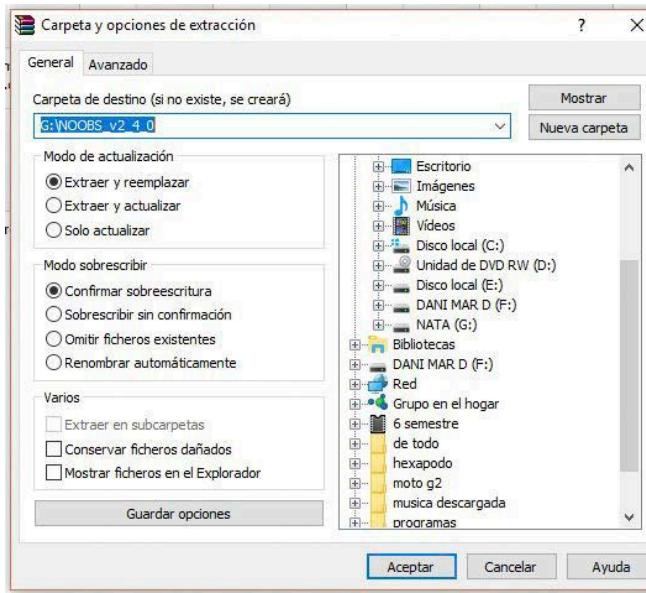
Figura 6  
Página de descarga del sistema operativo Raspberry Pi  
Fuente: Elaboración propia.

**Paso 2** En la carpeta de descargas, busque el archivo “NOOBS\_v2\_2\_0.zip”, selecciónelo y haga doble clic para descomprimirlo.



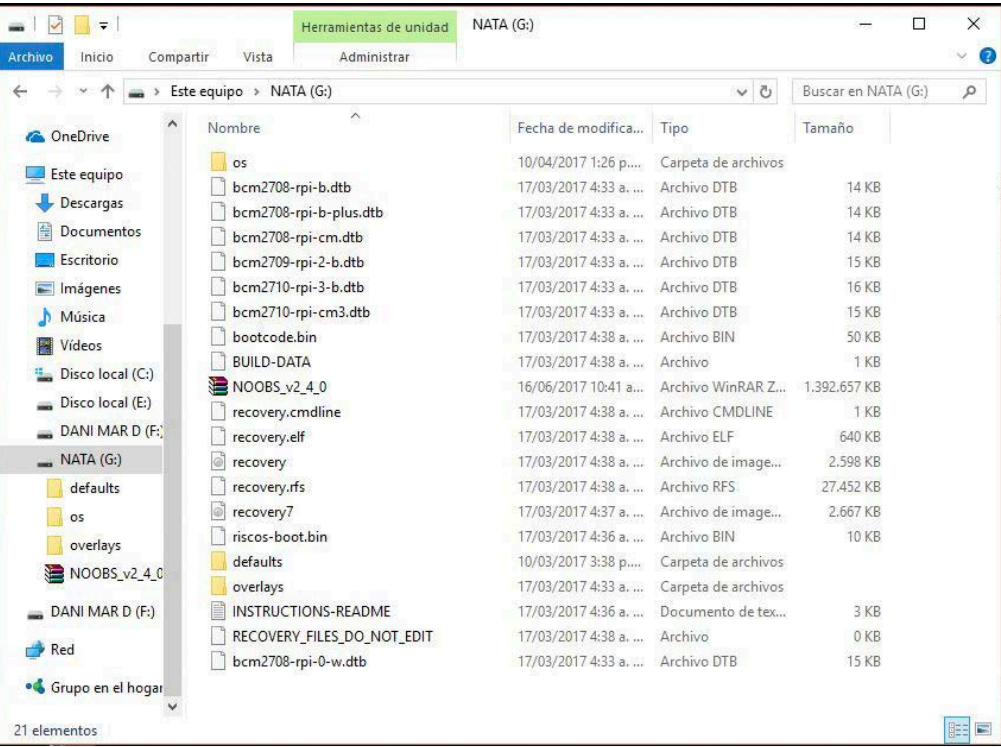
**Figura 7**  
Interfaz del explorador de Windows, carpeta de descargas  
Fuente: Elaboración propia.

**Paso 3** Descomprima NOOBS en la tarjeta micro SD.



**Figura 8**  
Ruta para descomprimir el sistema operativo Rapsberry Pi  
Fuente: Elaboración propia.

De esta forma, quedará listo el instalador con todos los paquetes que requiere para instalar el sistema operativo, paquete de oficina, *software* de programación y aplicativos accesorios.

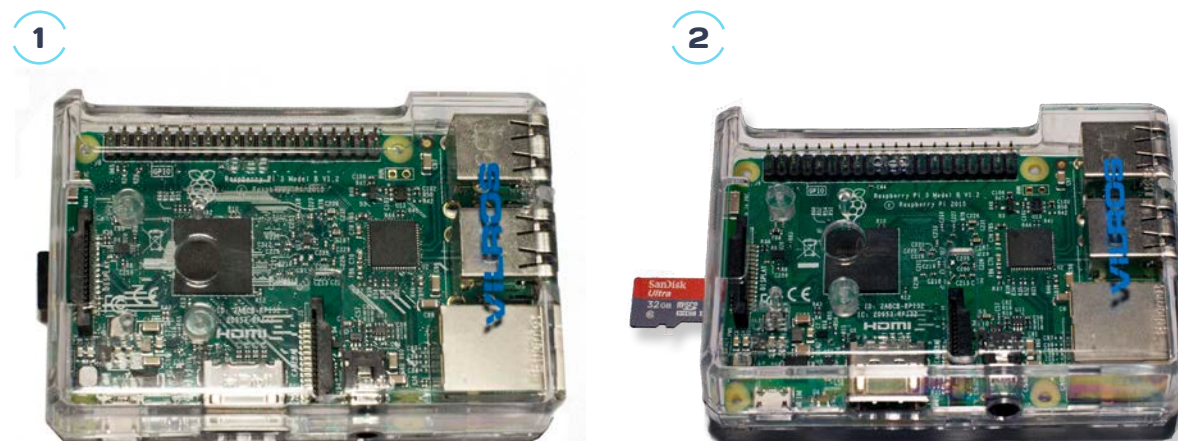


**Figura 9**  
Carpeta de instalación para Raspberry Pi  
Fuente: Elaboración propia.



### 1.3 Instalar el sistema operativo y el software aplicativo

- Paso 1 Inserte la tarjeta micro SD en la Raspberry Pi, conecte teclado, ratón, monitor, cable Ethernet y el cable de alimentación a 5V.

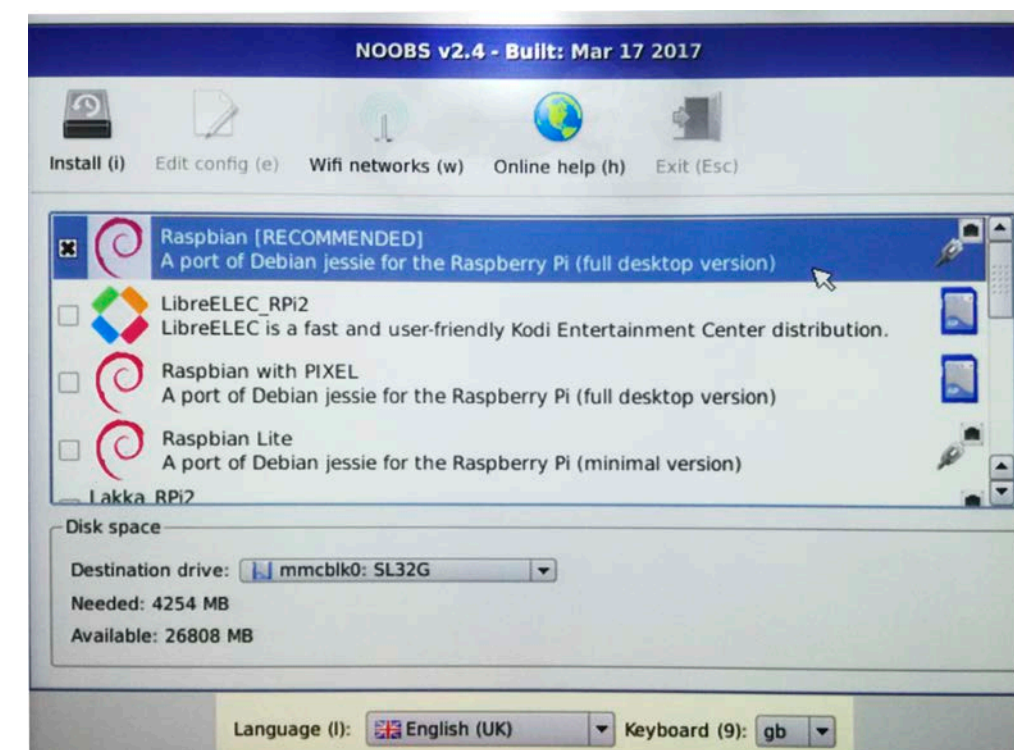


**Figura 10**

Instalación de la SD Card en el sistema embebido Raspberry Pi  
Fuente: Elaboración propia.

#### Paso 2

Al encender el microcomputador Raspberry Pi, encontrará el gestor de instalación en la tarjeta micro SD. Ejecútelo para iniciar la instalación del sistema operativo.



**Figura 11**

Interfaz de instalación de NOOBS para Raspberry Pi  
Fuente: Elaboración propia.

**Paso 3** La ventana inicial brinda la posibilidad de instalar varios sistemas operativos.

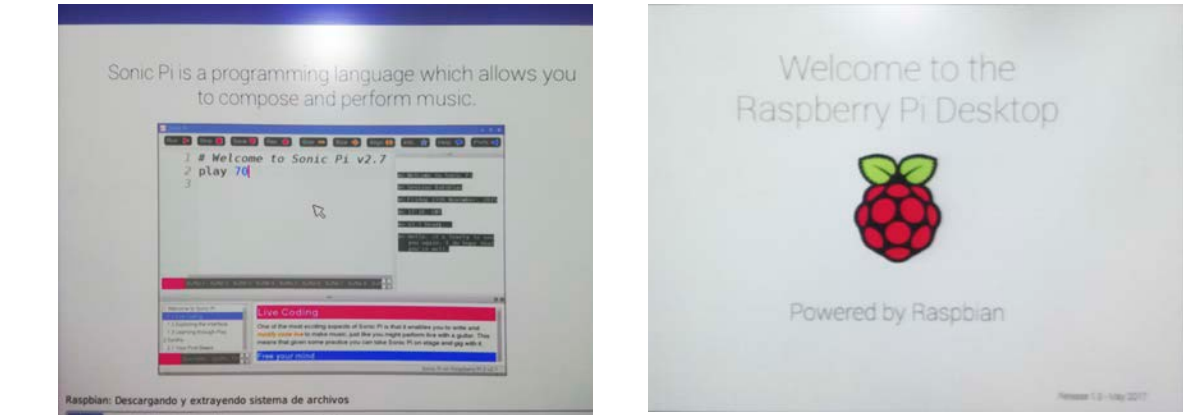
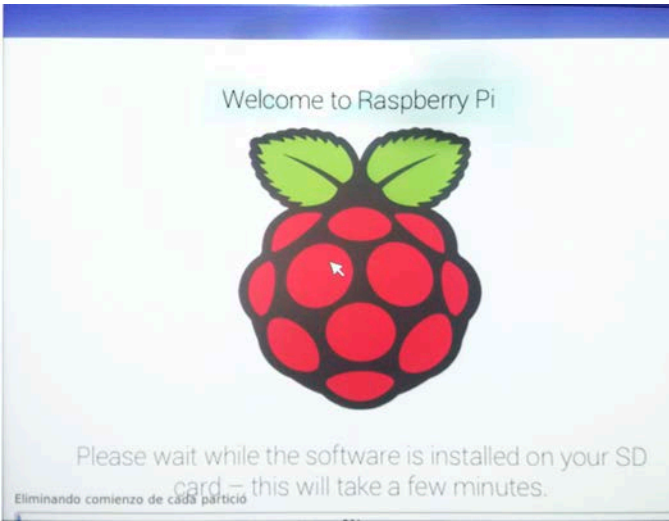
Seleccione Raspbian y haga clic sobre el ícono de instalar (parte superior izquierda de la ventana).

Aparecerán algunas ventanas de advertencia, léalas y haga clic en aceptar.

Espere a que se instale el SO Raspbian y todo el paquete de aplicativos (tomará alrededor de una hora).

Figura 12

Procedimiento de instalación del sistema operativo Raspbian  
Fuente: Elaboración propia.



**Paso 4** Al terminar la instalación, ingrese a Pixel, el entorno gráfico de Raspbian.

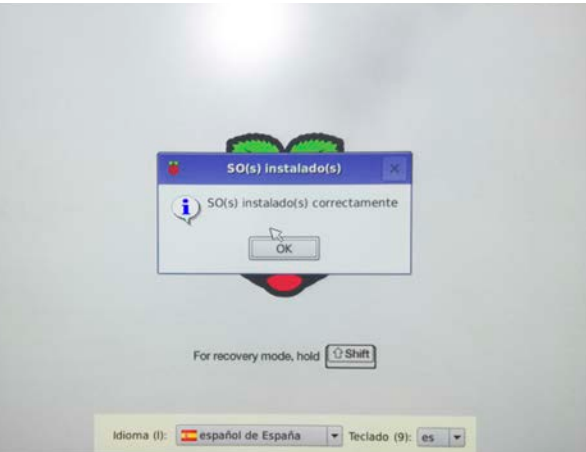
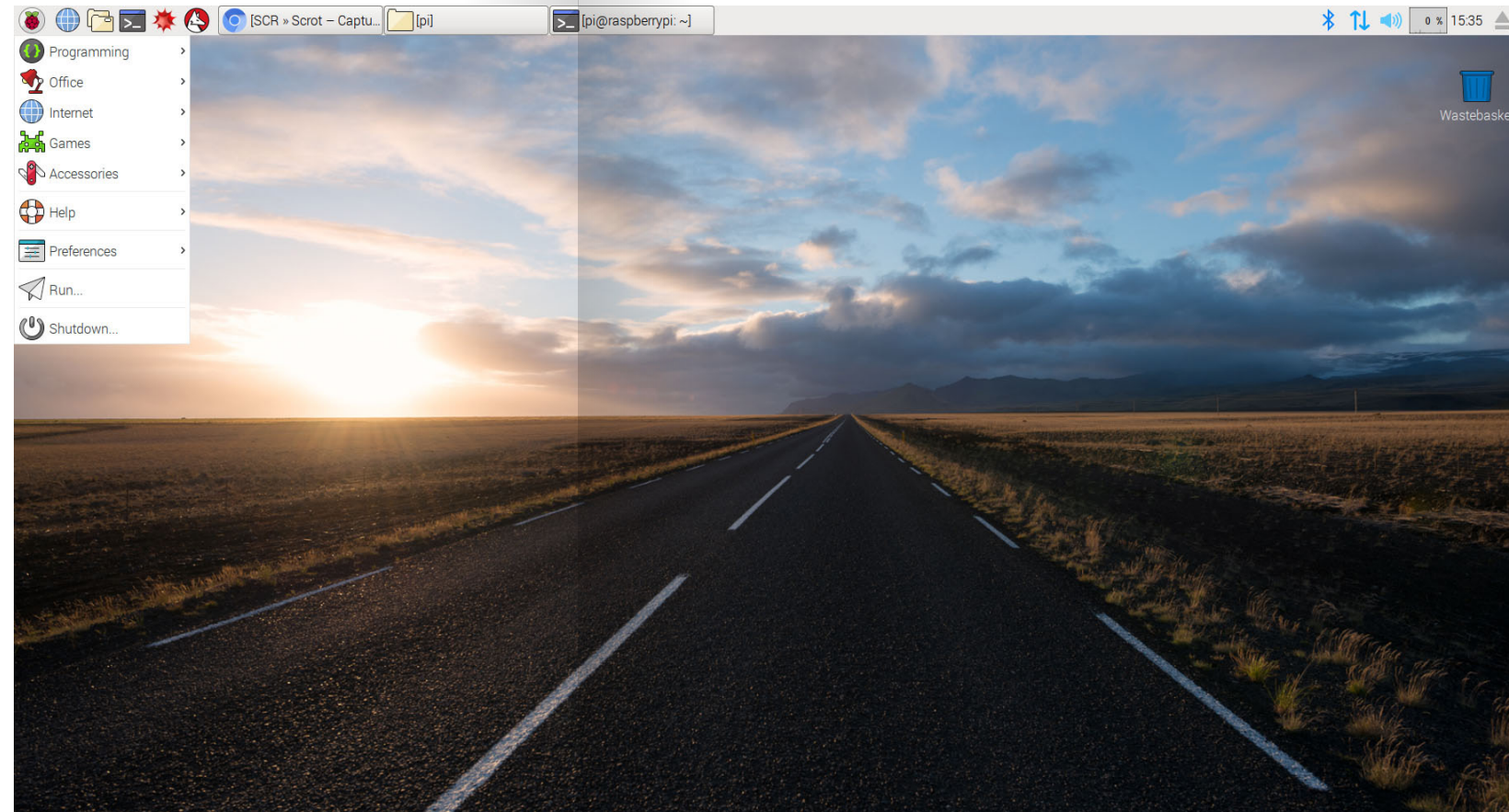


Figura 13

Finalización de la instalación de Raspbian en la SD Card  
Fuente: Elaboración propia.

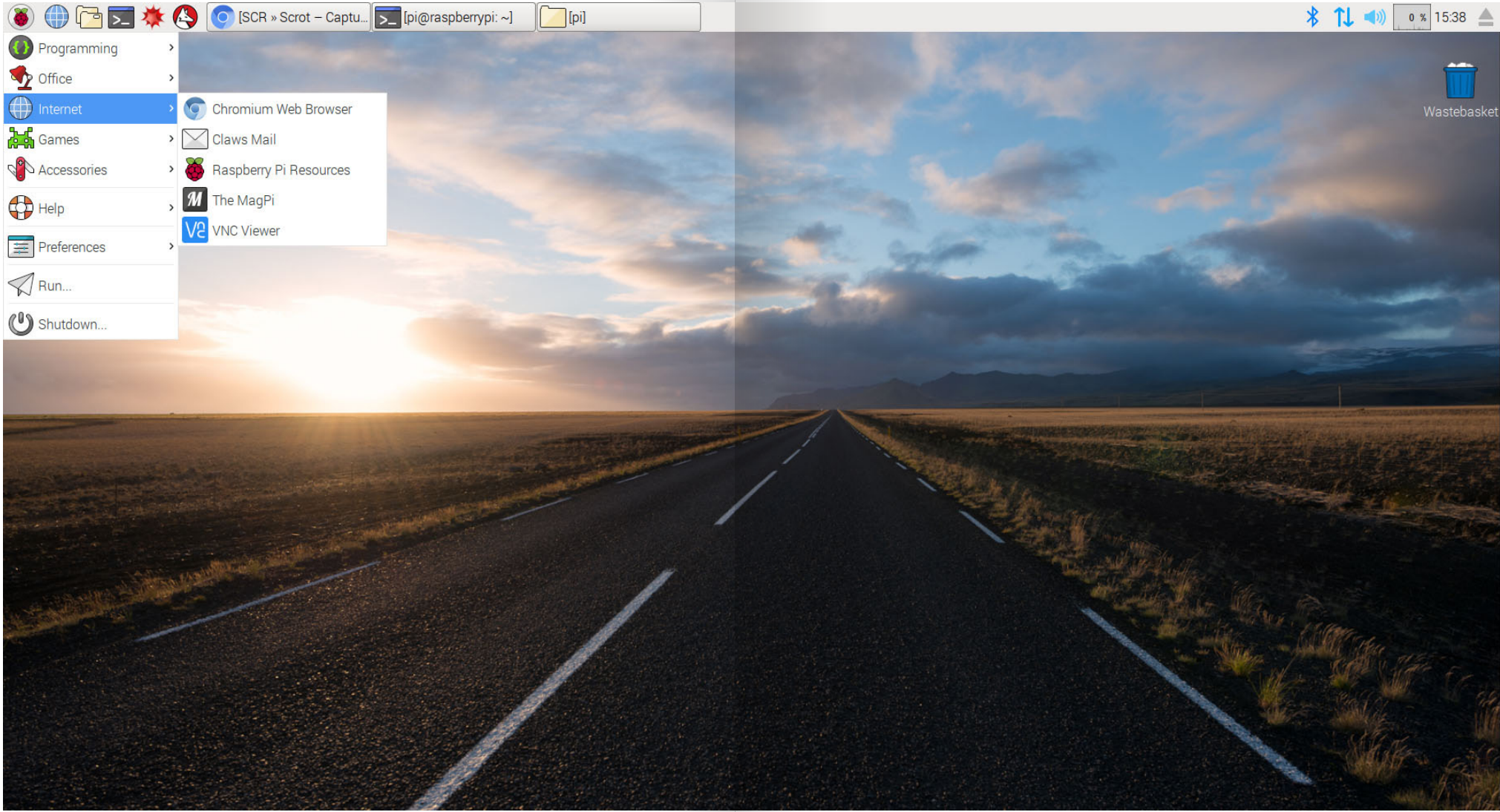
La barra de tareas y programas está ubicada en la parte superior; en el extremo izquierdo se encuentra el botón “Raspberry” con el menú de programas.

Estos se encuentran organizados en aplicativos de programación, oficina, Internet, juegos, accesorios, *help* y preferencias.

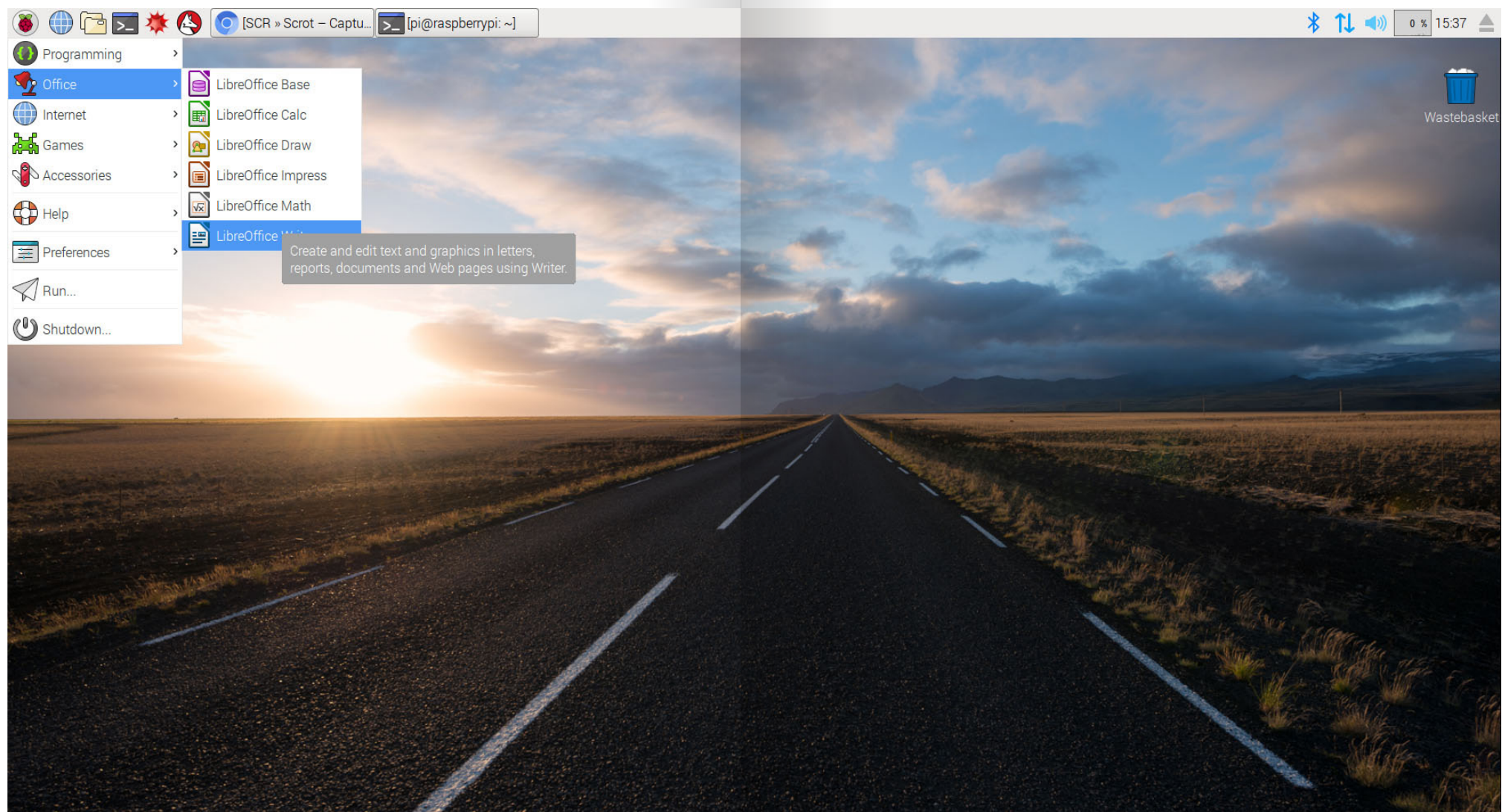


**Figura 14**  
Descripción del escritorio de trabajo de Raspbian  
Fuente: Elaboración propia.

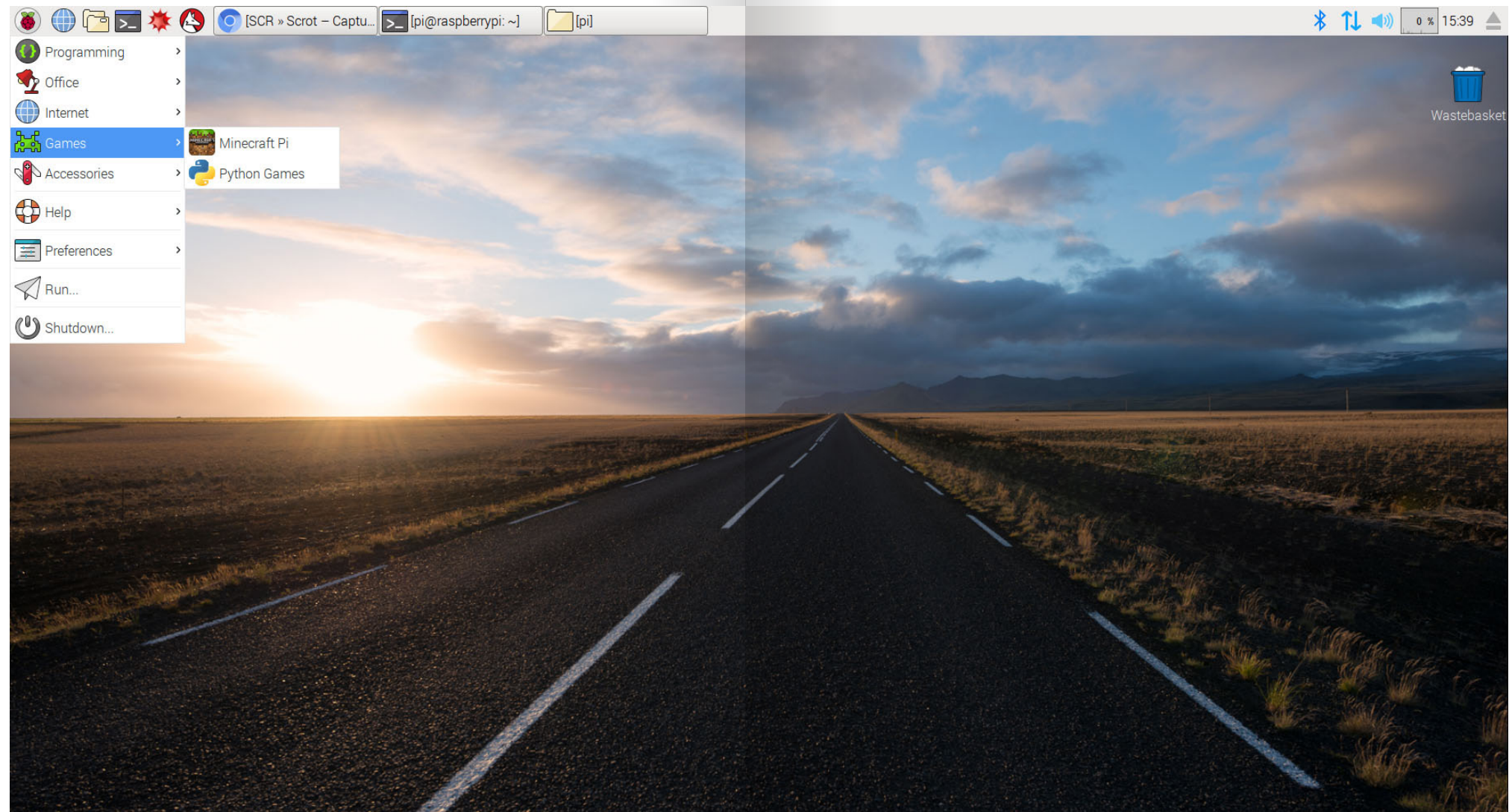










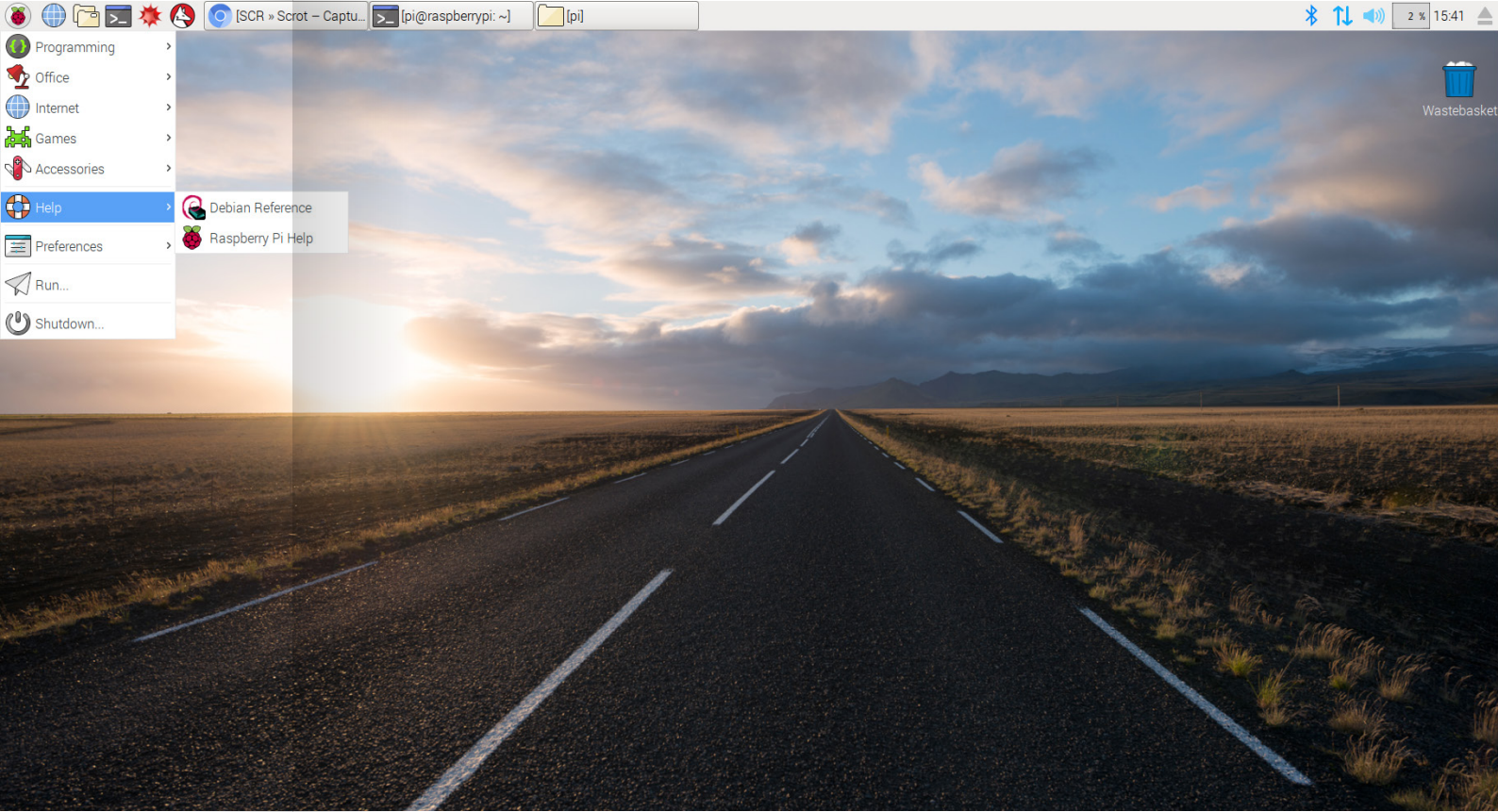


### 1.4 Cambiar la configuración del teclado

● **Nota aclaratoria:** Si aparecen símbolos extraños cuando se escribe con el teclado, se requiere realizar la configuración del teclado para el entorno gráfico. En sistemas Linux es frecuente hablar del “terminal” o la “consola” para referirse al entorno de comandos, el entorno no gráfico o el entorno base de Linux.

● Paso 1      **Activar una ventana de terminal**

Para activar una ventana de terminal, seleccione el icono >\_ en la parte izquierda de la barra de tareas. En adelante, los comandos requeridos en una ventana de terminal tendrán formato de letra blanca sobre fondo negro.



**Figura 15**

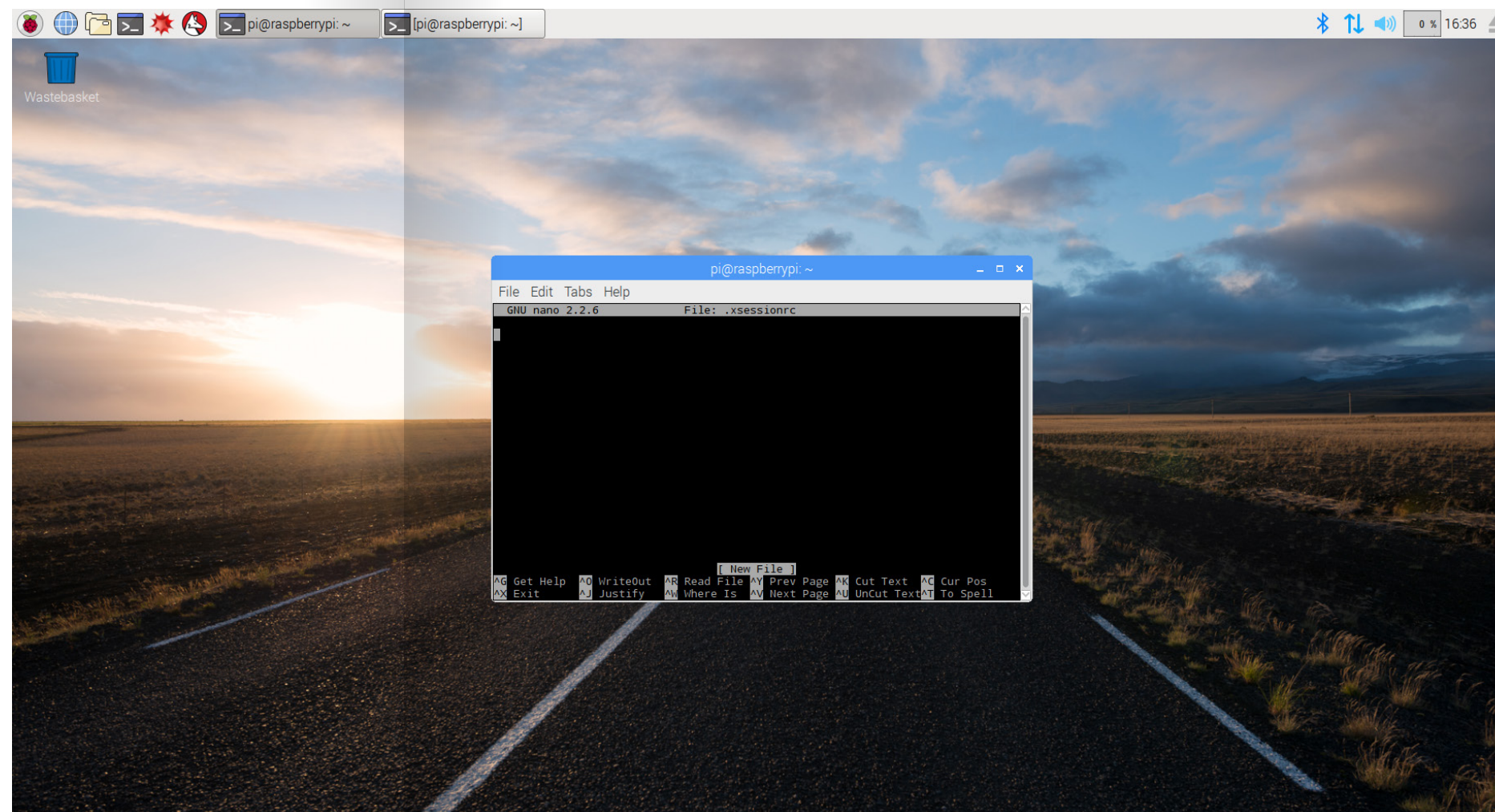
Panel de configuración de los periféricos de Raspbian  
Fuente: Elaboración propia.



## Paso 2

Debe mostrar la ubicación actual dentro del sistema de carpetas, que inicialmente será el directorio de usuario, presentando el nombre del directorio:

**pi@raspberrypi: ~ \$**



**Figura 16**

Terminal del sistema operativo Raspbian  
Fuente: Elaboración propia.



**Paso 3** Cree el archivo de configuración del entorno gráfico y ábralo para editarlo con Nano (un editor de texto en el entorno de comandos). Digite:

**sudo nano .xsessionrc**

Dentro del archivo, escriba:

**setxkbmap es**

Guarde y salga (Ctrl+o, enter, Ctrl+x)

## 2. Configuraciones adicionales

### 2.1 Instalar utilidades para manipular archivos .xls con Python

**Nota:** se requiere acceso a Internet.

En una ventana de terminal descargue las librerías e instálelas digitando:

**sudo pip install xlwt**

**sudo pip install xlrd**

**sudo pip install xlutils**

*pip* es una utilidad de instalación.

xlwt contiene rutinas de escritura.

xlrd contiene rutinas de lectura.

xlutils contiene la rutina de manipulación de archivos.

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo pip install xlwt
Downloading/unpacking xlwt
  Downloading xlwt-1.2.0-py2.py3-none-any.whl (99kB): 99kB downloaded
Installing collected packages: xlwt
Successfully installed xlwt
Cleaning up...
pi@raspberrypi:~ $ sudo pip install xlrd
Downloading/unpacking xlrd
  Downloading xlrd-1.0.0.tar.gz (2.6MB): 2.6MB downloaded
  Running setup.py (path:/tmp/pip-build-LmzJXz/xlrd/setup.py) egg_info for packa
ge xlrd
    warning: no files found matching 'README.html'
Installing collected packages: xlrd
  Running setup.py install for xlrd
    changing mode of build/scripts-2.7/runxlrd.py from 644 to 755
    warning: no files found matching 'README.html'
    changing mode of /usr/local/bin/runxlrd.py to 755
Successfully installed xlrd
Cleaning up...
pi@raspberrypi:~ $ sudo pip install xlutils
Downloading/unpacking xlutils
  Downloading xlutils-2.0.0-py2.py3-none-any.whl (55kB): 55kB downloaded
Requirement already satisfied (use --upgrade to upgrade): xlwt>=0.7.4 in /usr/lo
cal/lib/python2.7/dist-packages (from xlutils)
Requirement already satisfied (use --upgrade to upgrade): xlrd>=0.7.2 in /usr/lo
cal/lib/python2.7/dist-packages (from xlutils)
Installing collected packages: xlutils
Successfully installed xlutils
Cleaning up...
pi@raspberrypi:~ $
```

**Figura 17**

Instalación de paquetes desde el terminal del sistema operativo  
 Fuente: Elaboración propia.

NOTA: Para mayor información y ejemplos, consulte la página:



<https://ubuntulife.wordpress.com/2011/09/25/crear-leer-y-modificar-una-hoja-excel-con-python/>

50

Ejemplo:

```
#Código en Python que crea archivo, inserta hoja "Datos", inserta información y guarda el archivo .xls.
import xlwt                                     #Importa librería de escritura
import time                                    #Importa librería de manipulación de tiempo y fechas
wb = xlwt.Workbook()                           #Crea nuevo espacio de trabajo
ws = wb.add_sheet('Datos')                     #Inserta nueva hoja llamada "Datos" y habilita escritura

ws.write(0, 0, 'FECHA', style0)                 ('Datos',cell_overwrite_ok=True)
ws.write(1, 0, datetime.now(), style1)         #En la celda A1 (fila0,col0) escribe "FECHA"
ws.write(2, 0, 4)                              #En la celda A2 escribe la fecha actual
ws.write(2, 1, 1)
ws.write(2, 2, xlwt.Formula("A3+B3"))
wb.save('Ejemplo.xls')                         #En la celda B3 escribe "1"

#Guarda el archivo con nombre "Ejemplo.xls"
```

## 2.2 Configuración para subir archivos a Dropbox desde Python

### Paso 1

Ingresa a la página de Dropbox para desarrolladores: <https://www.dropbox.com/developers> e ingresa a la cuenta de usuario creada previamente. Para nuestro caso:

Nombre de usuario: **Proydom@yahoo.com**

Clave: **rasp2017.**

Seleccione el cuadro *Read our docs* para buscar información de la instalación en algún dispositivo de cómputo.

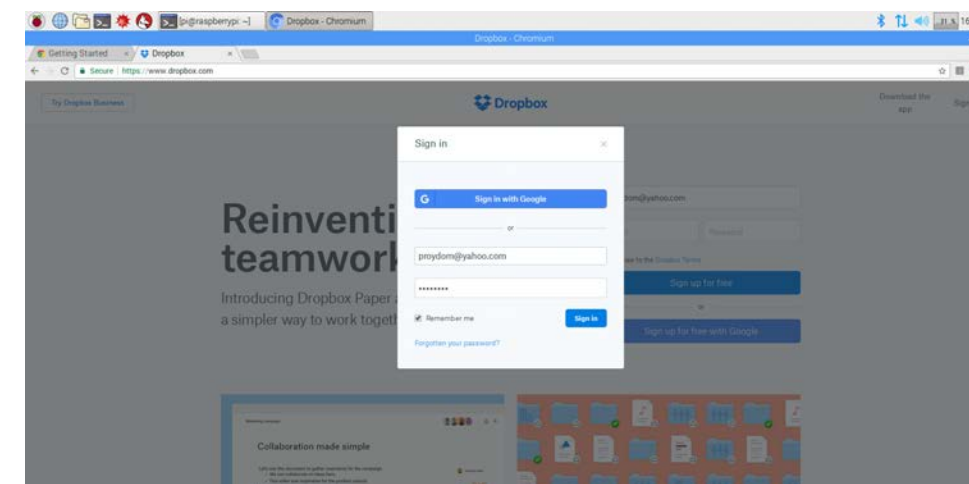


Figura 18

Configuración para subir archivos a Dropbox en modo programador  
Fuente: Elaboración propia.

### Paso 2

Seleccione la plataforma con la cual va a programar. En este caso, el círculo "Python".

51

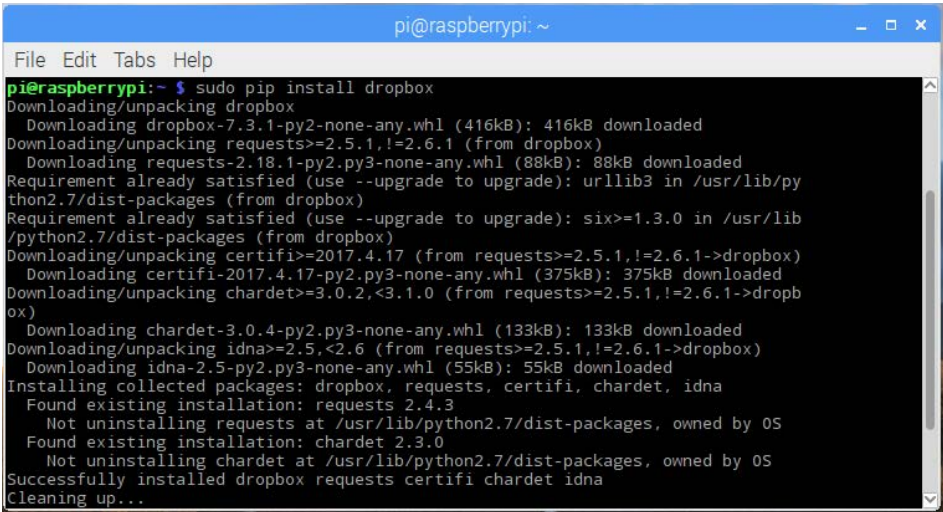
**Paso 3** Para descargar e instalar el SDK, haga clic en la pestaña *Install*, en la sección *Install the Python SDK*, y use el siguiente comando en una ventana de terminal:

**Sudo pip install dropbox**

[www.dropbox.com/develo](http://www.dropbox.com/develo)

**Paso 4** En el código Python que se está diseñando, importe la librería Dropbox, incluyendo esta línea en la cabecera:

**import dropbox**



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo pip install dropbox  
Downloading/unpacking dropbox  
  Downloading dropbox-7.3.1-py2-none-any.whl (416kB) downloaded  
Downloading/unpacking requests>=2.5.1,!<2.6.1 (from dropbox)  
  Downloading requests-2.18.1-py2.py3-none-any.whl (88kB) downloaded  
Requirement already satisfied (use --upgrade to upgrade): urllib3 in /usr/lib/py  
thon2.7/dist-packages (from dropbox)  
Requirement already satisfied (use --upgrade to upgrade): six>=1.3.0 in /usr/lib  
/python2.7/dist-packages (from dropbox)  
Downloading/unpacking certifi>=2017.4.17 (from requests>=2.5.1,!<2.6.1->dropbox)  
  Downloading certifi-2017.4.17-py2.py3-none-any.whl (375kB) downloaded  
Downloading/unpacking chardet>=3.0.2,<3.1.0 (from requests>=2.5.1,!<2.6.1->dropb  
ox)  
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB) downloaded  
Downloading/unpacking idna>=2.5,<2.6 (from requests>=2.5.1,!<2.6.1->dropbox)  
  Downloading idna-2.5-py2.py3-none-any.whl (55kB) downloaded  
Installing collected packages: dropbox, requests, certifi, chardet, idna  
  Found existing installation: requests 2.4.3  
    Not uninstalling requests at /usr/lib/python2.7/dist-packages, owned by OS  
  Found existing installation: chardet 2.3.0  
    Not uninstalling chardet at /usr/lib/python2.7/dist-packages, owned by OS  
Successfully installed dropbox requests certifi chardet idna  
Cleaning up...
```

**Figura 19**

Instalación del paquete de Dropbox en Raspbian  
Fuente: Elaboración propia.

**Paso 5** Para acceder desde Python a los archivos de Dropbox en la nube sin requerir autenticación en cada proceso, se requiere un token de acceso (código de acceso directo).

Para obtenerlo, vaya a la página de Dropbox para desarrolladores y escoja el cuadro *Create your app*.

**Paso 6** En la página *Create a new app on the Dropbox Plat-*  
*form*, en el numeral 1, *haga clic en Dropbox API*.

En el numeral 2, haga clic en *Full Dropbox- Access*  
*to all files and folders in a user's Dropbox*.

En el numeral 3, dé un nombre a la aplicación (no  
acepta nombres repetidos).

En la nueva ventana de la aplicación, pestaña *Set-*  
*tings*, busque el campo *Generated access token* y  
seleccione el botón *Generate*.

Como resultado, Dropbox proporciona un *Access*  
*Token* de ingreso exclusivo.

En este caso, para la aplicación *Sensor\_Hum\_Tem.py*  
generó el token:

**XE-R9TnDEEEAAAAAAAAAAC6UFxNuopczngi-  
tvVQauZ4A3iY20Z6Lx6kiRa0s9Pa4e**

De esta manera, cuando Python requiera acceder a Dropbox, tendrá que hacerlo mediante la estructura:

```
Dbx=Dropbox.Dropbox(XE-R9Tn-DEEAAAAAAAAAAC6UFxNuopczngi-tvVQauZ4A3iY20Z6Lx6kiRa0s9Pa4e)
```

Ahora dbx tiene las propiedades:

```
dbx.files_upload(EstructuradeDatos,'ruta en Dropbox', modo, mute)
#Para subir archivos a Dropbox

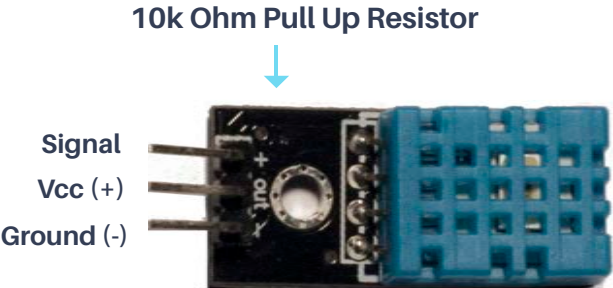
dbx.files_download_to_file('Archivo-Destino.xls', 'ruta en Dropbox')
#Para descargar archivos
```

### 3. Aplicación: subir datos a la nube (medición de temperatura/humedad)

- **Materiales**
  - \* Sistema microcomputador Raspberry Pi 3 (con periféricos conectados)
  - \* Sensor de temperatura y humedad relativa DHT11
  - \* Cables de conexión rápida

### 3.1 El Sensor DHT11

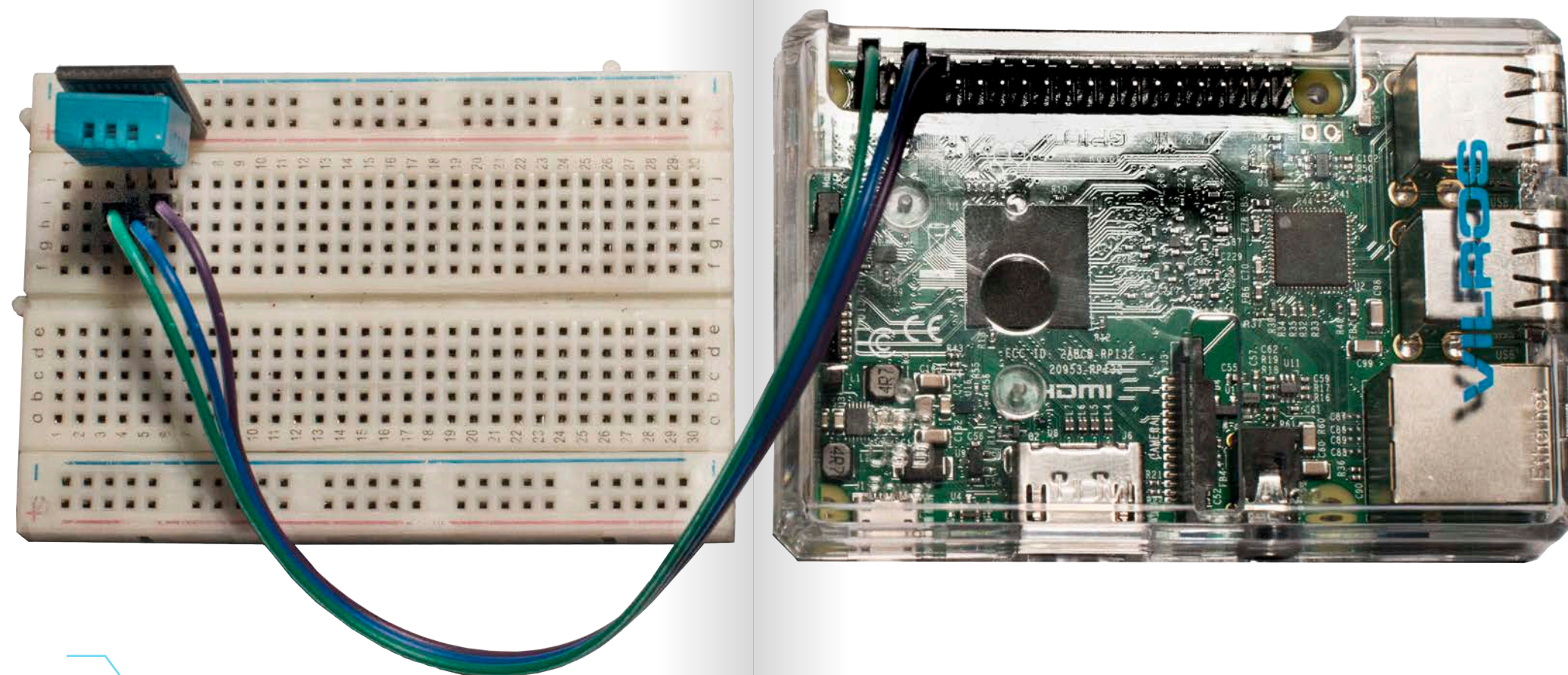
● **Presentación:** Este sensor (Figura 20) sirve para medir tanto la humedad relativa como la temperatura. Tiene resolución de 1% para la humedad relativa y de 1°C para la temperatura. Se caracteriza por tener la señal digital calibrada mediante un microcontrolador de 8 bits integrado, lo que asegura alta calidad y fiabilidad en las lecturas. Está constituido por dos sensores resistivos: NTC y humedad. Rango de medición de humedad: 20% - 95% Rango de medición de temperatura: 0°C - 50°C. El protocolo de comunicación es serial, a través de un único hilo (protocolo 1-wire) y con capacidad para transmitir la señal hasta 20 metros de distancia.



**Figura 20**  
Sensor de temperatura y humedad DHT11  
Fuente: Elaboración propia.



Paso 1 Conecte la Raspberry Pi siguiendo la Figura 21.



**Figura 21**

Conexión de Raspberry Pi con sensor DHT11  
Fuente: Elaboración propia.

## Paso 2

Si se tiene en cuenta que esta tarjeta tiene al menos dos modos de numeración para el puerto GPIO (entradas/salidas digitales), la conexión se debe hacer siguiendo la numeración física de los pines.

- Terminal Negativo del sensor – Pin 6 de Raspberry Pi
- Terminal Positivo del sensor – Pin 2 de Raspberry Pi
- Terminal Data del sensor – Pin 7 de Raspberry Pi



**Figura 22**

Numeración de los pines de Raspberry Pi  
 Fuente: Elaboración propia.

## Paso 3

### Instalar la librería para el sensor DHT11

Se utiliza una librería de la casa Adafruit para este tipo de sensores.

En una ventana de terminal, ingrese el comando para copiar los archivos:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

Ingresa al directorio recién copiado:

```
cd Adafruit_Python_DHT
```

Instale la librería con los comandos:

```
sudo apt-get install build-essential python-dev
```

Y a continuación digite:

```
sudo python setup.py install
```

## Paso 4

### Programa en Python

Para escribir el código en el entorno gráfico de Raspbian, seleccione el botón de inicio *Raspberrry*, menú *Programación* y seleccione:

*Python 2 (IDLE)*.

Al iniciarse la ventana de edición de Python, seleccione del menú *File* la opción *New File*.

En dicha ventana, ingrese el siguiente código:

```
#Programa que lee datos de temperatura y humedad desde un sensor DHT11,
#guarda los datos en un archivo .xls y lo sube a la nube a una carpeta de Dropbox

#!/usr/bin/python
import sys
import xlwt
import time
import Adafruit_DHT
import dropbox

#Librería para modificar archivos .xls
#Librería para manipulación de variables temporales
#Librería para sensor DHT11
#Librería para Dropbox
```

```
MaxLect=60                                #Número máximo de lecturas
Lect=1

wb = xlwt.Workbook()                      #Carga un nuevo archivo
ws = wb.add_sheet('Datos',cell_overwrite_ok=True) #Crea una nueva hoja llamada "Datos"
ws.write(0, 0, 'ID')                      #Escribe ID en la celda A1
ws.write(0, 1, 'TEMPERATURA')            #Escribe TEMPERATURA en la celda B1
ws.write(0, 2, 'HUMEDAD')

while (Lect <= MaxLect):
    humidity, temperature = Adafruit_DHT.read_retry(11, 4) #Lee datos de humedad y temperatura
    ws.write(Lect, 0, Lect)                  #Genera consecutivo
    ws.write(Lect, 1, temperature)          #Escribe datos
    ws.write(Lect, 2, humidity)

    time.sleep(10)                          #Retardo
    Lect = Lect + 1

wb.save('Hum_Tem.xls')                    #Salva datos en el archivo 'Hum_Tem.xls'

dbx = dropbox.Dropbox('XE-R9TnDEEAAAAAAAAAAAC6UFxNuopcngitvVQauZ4A3iY20Z6Lx6ki-
Ra0s9Pa4e') #Carga Token
fi = open('Hum_Tem.xls', 'rb')             #Abre archive local
str = fi.read()                           #Crea estructura con los datos
dbx.files_upload(str, '/Hum_Tem.xls', dropbox.files.WriteMode.overwrite, mute = True) #Crea archi-
vo en Dropbox
fi.close()                                #Cierra archivo local
```

Guardar con nombre: sensor\_hum\_tem.py

En una ventana de terminal ingrese el comando:

**Sudo python sensor\_hum\_tem.py**

Verifique que en la cuenta de Dropbox se haya creado el archivo "Hum\_Tem.xls".

#### 4. Aplicación: subir datos a la nube (detección de presencia de personas)

##### ● Materiales

- \* Sistema microcomputador Raspberry Pi 3 (con periféricos conectados)
- \* Sensor de PIR HC-SR501
- \* Cables de conexión rápida

##### ● El sensor HC-SR501

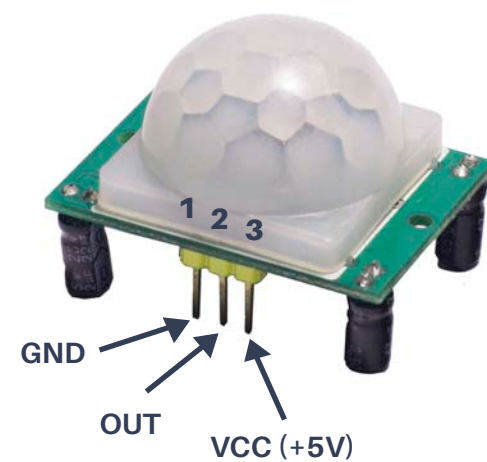
##### ● Presentación:

Debido a que todos los seres vivos desprenden calor en forma de radiación infrarroja, y lo mismo ocurre con los automóviles y cualquier otra maquinaria, se puede detectar su presencia con este tipo de sensores PIR (Figura 23). Estos sensores detectan cambios en la radiación infrarroja que reciben, activando una salida de control.



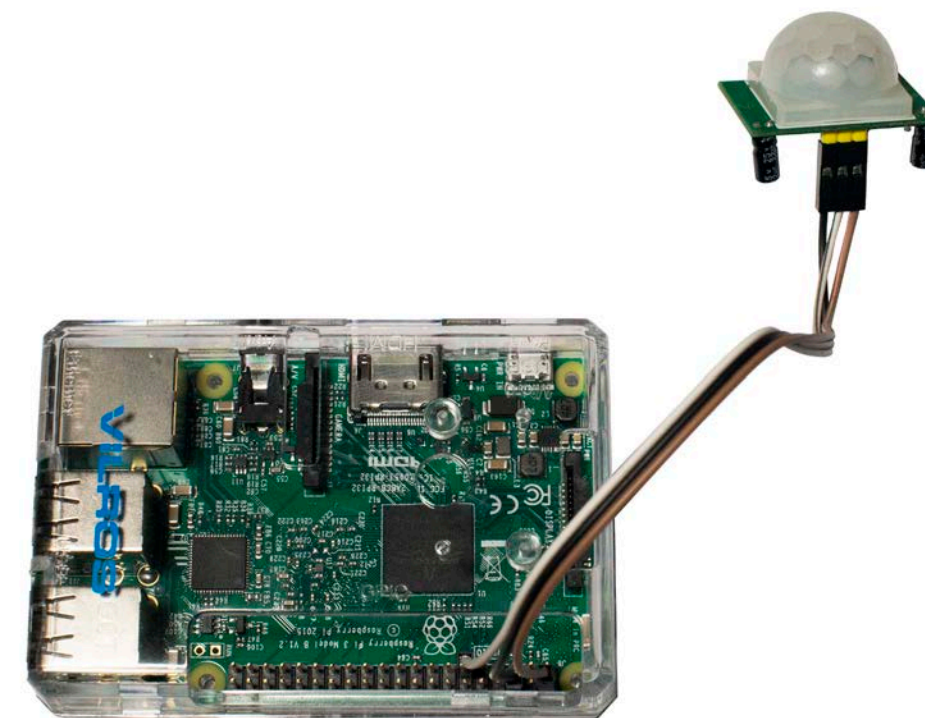
Los PIR detectan el movimiento comparando el estado de dos ventanas IR. Los objetos estáticos frente al sensor generan patrones similares en las dos ventanas, mientras que los objetos en movimiento generan patrones diferentes si se mueven más rápido. Poseen una lente Fresnell para mejorar el ángulo de detección, al dividir el área de detección en sectores con diferentes lentes.

Se alimentan con 5V y la señal de salida tiene nivel de 3.3V.



**Figura 23**  
Sensor de presencia HC-SR501  
Fuente: Elaboración propia.

**Paso 1** Conectar a la Raspberry Pi siguiendo el gráfico de ejemplo (ibeyonde, 2016).



**Figura 24**  
Esquema de conexión del sensor de presencia con Raspberry Pi  
Fuente: Elaboración propia.



## Paso 2

Debido a que esta tarjeta tiene al menos dos modos de numeración para el puerto GPIO (entradas/salidas digitales), la conexión se debe hacer siguiendo la numeración física de los pines.

- Terminal Negativo del sensor – Pin 6 de Raspberry Pi
- Terminal Positivo del sensor – Pin 2 de Raspberry Pi
- Terminal Data del sensor – Pin 7 de Raspberry Pi



**Figura 25**

Esquema de pines GPIO para conexión con la Raspsberry Pi  
 Fuente.: Elaboración propia.

## 4.2 Instalar librería para el puerto GPIO

### Paso 3

Se utiliza una librería genérica para la tarjeta Raspberry Pi.

En una ventana de terminal, ingrese el comando para copiar los archivos (Internet):

**git clone git://git.drogon.net/wiringPi**

Ingresa al directorio recién copiado:

**cd wiringPi**

Compile el directorio:

**./build**

Compruebe el funcionamiento leyendo entradas del GPIO:

**gpio readall**

## Paso 4

### Programa en Python

Para escribir el código en el entorno gráfico de Raspbian, seleccione en el botón de inicio *Raspbian*, menú *Programación*:

*Python 2 (IDLE)*

Se inicia una ventana de edición de Python, seleccione del menú *File* la opción *New File*.

En dicha ventana ingrese el siguiente código:

```
#Programa que lee datos de un sensor PIR HC-SR501, guarda los datos en un archivo .xls
#y lo sube a la nube a una carpeta de Dropbox
#!/usr/bin/python
import sys
import xlwt
import RPi.GPIO as GPIO
import time
import dropbox

#Librería genérica
#Librería para modificar archivos .xls
#Librería para acceder a puertos I/O
#Librería para manipulación de variables temporales
#Librería para Dropbox
```

```

pir_sensor = 11          #Asigna números de pines
led = 7

GPIO.setmode(GPIO.BOARD)
GPIO.setup(led,GPIO.OUT)  #Configura pin como salida
GPIO.setup(pir_sensor, GPIO.IN) #Configura pin como entrada
current_state = 0

MaxLect=60
Lect=1

wb = xlwt.Workbook()      #Bloque de inicio de archive .xls
ws = wb.add_sheet('Datos',cell_overwrite_ok=True)
ws.write(0, 0, 'ID')
ws.write(0, 1, 'HORA')
ws.write(0, 2, 'MOVIMIENTO')

while (Lect <= MaxLect):
    time.sleep(0.1)
    current_state = GPIO.input(pir_sensor)          #Lee el estado del sensor
    if current_state == 1:                          #Si está activado...
        print "Movimiento detectado" #Genera aviso por pantalla...
        print "", time.strftime("%X")
        GPIO.output(led,True)                #Enciende led
        ws.write(Lect, 0, Lect)                #Y guarda información de estado activo (movimiento)
        ws.write(Lect, 1, time.strftime("%X"))
        ws.write(Lect, 2, 1)
    else:
        GPIO.output(led,False)                #Cuando no está activado guarda
        ws.write(Lect, 0, Lect)                #Información de estado desactivado
        ws.write(Lect, 1, time.strftime("%X"))
        ws.write(Lect, 2, 0)
  
```

```

time.sleep(0.5)          #Retardo
Lect = Lect + 1

GPIO.cleanup()           #Desactiva Puerto de I/O
wb.save('Movimiento.xls') #Guarda datos en archive .xls

dbx = dropbox.Dropbox('XE-R9TnDEEAAAAAAAAAAAC6UFxNuopcngitvVQauZ4A3iY20Z6Lx6ki-
Ra0s9Pa4e')
fi = open ('Movimiento.xls', 'rb')
str = fi.read()
dbx.files_upload(str, '/Movimiento.xls', dropbox.files.WriteMode.overwrite, mute = True)
#Sube archive a Dropbox

fi.close()
  
```

Guarde con el nombre sensor\_PIR.py  
 En una ventana de terminal, ingrese el comando:

**Sudo python sensor\_PIR.py**

Verifique que en la cuenta de Dropbox se haya creado el archivo "Movimiento.xls".

**Para mayor información, consulte las siguientes páginas:**

<https://www.raspberrypi.org/>  
<http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/>  
<https://github.com/dropbox/dropbox-sdk-python/blob/master/example/backup-and-restore/backup-and-restore-example.py>  
<http://fpaez.com/sensor-de-movimiento-infrarojo-hc-sr501/>

Usted puede acceder a cualquier sistema de almacenamiento en la nube como Dropbox o OneDrive para consultar los datos. En este caso se utilizó Dropbox.



# pcDuino

## 1. Cómo hacer que funcione pcDuino

### Presentación

pcDuino es una placa de ordenador que puede ejecutar una versión de Linux, Ubuntu y también puede correr en Android.

Esta placa incluye procesador dual core M20 a 1Ghz, 1Gbyte de RAM, a 32 bits. Incluye WIFI, Ethernet, lector de tarjetas SD, 4 GBytes de memoria Flash para arrancar el sistema operativo, procesador gráfico Mali, salida de video HDMI, un conector SATA para discos duros y un receptor de infrarrojos.

### Materiales

- \* Microcomputador PCDuino
- \* Memoria micro SD ( $\geq 8$  Gb)
- \* Fuente de alimentación 5V, 1A con conector micro USB
- \* Multiplicador Hub USB
- \* Teclado USB
- \* Monitor (conexión HDMI)
- \* Ratón USB (opcional)
- \* Cable Ethernet

**Figura 26**  
Diagrama de configuración de pcDuino  
Fuente: Elaboración propia.

**J11**

- 1 GPIO0 / UART2 RX
- 2 GPIO1 / UART2 RX
- 3 GPIO2
- 4 GPIO3 / PWM3
- 5 GPIO4
- 6 GPIO5 / PWM5
- 7 GPIO6 / PWM6
- 8 GPIO7

**J8**

- 1 GPIO8
- 2 GPIO9 / PWM9
- 3 SPI\_SS (Subordinate / Chip Select) /GPIO10 / PWM10
- 4 SPIO\_MISO (Master Output, Subordinate Input) /GPIO11 / PWM11
- 5 SPIO\_MISO (Master Input, Subordinate Output) /GPIO12
- 6 SP10\_CLK (Clock) / GPIO13
- 7 GND
- 8 AREF
- 9 TW12\_SDA (Data)
- 10 TW12\_SCL (Clock)

**P6**

- 1 SP11\_MISO (Master Input, Subordinate Output)
- 2 +5V DC
- 3 SP11\_CLK (Clock)
- 4 SP11\_MOSI (Master Output, Subordinate Input)
- 5 RESET
- 6 GND

**P7**

- 1 SP110\_MISO (Master Input, Subordinate Output)
- 2 +5V DC
- 3 SP10\_CLK (Clock)
- 4 SP10\_MOSI (Master Output, Subordinate Input)
- 5 RESET
- 6 GND

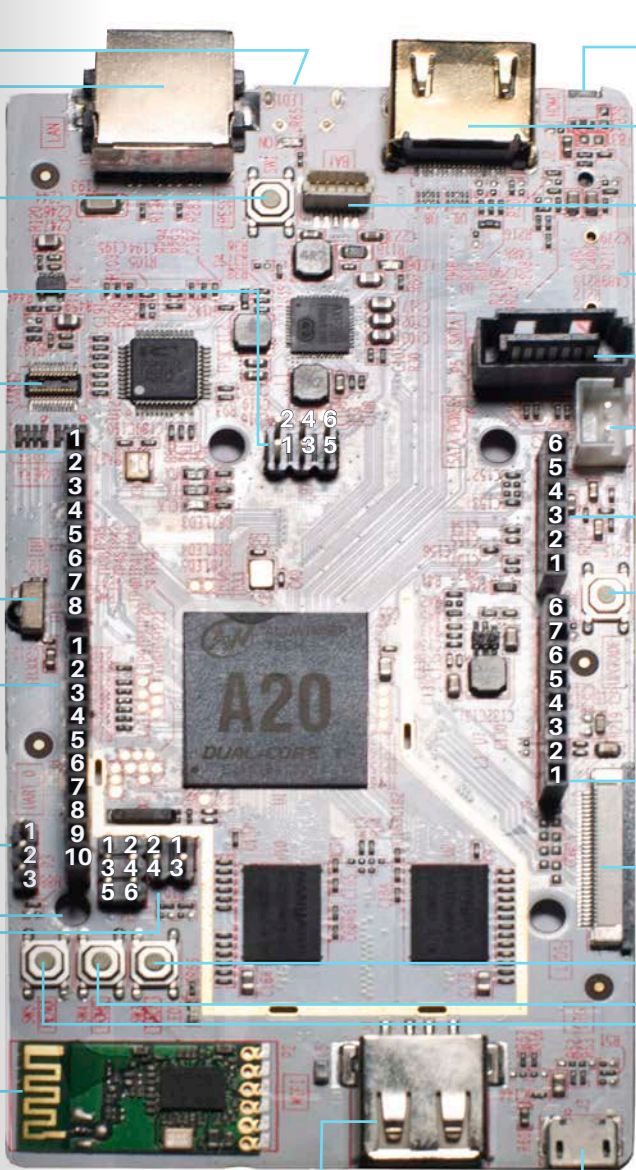
**P3 (Debug Port)**

- 1 TX ( Square Pad)
- 2 GND
- 3 RX

**P10**

- 1 GPIO14
- 2 GPIO15
- 3 GPIO16
- 4 GPIO17

- J11** DC Power IN
- J13** Network RJ45
- SW1** Reset
- P7** SPI0 Port
- CON15** Camera
- J11** Digital I/O
- U16** IR
- J8** Digital I/O
- P3** Debug Port
- P6** SPII Port
- P10** GPIO
- P2** Wireless Connection Module Interface
- J4** Host USB (USB2.0)



- CN8** Audio
- J2** HDMI
- J7** Battery Connector
- J5** TF\_card(bottom)
- P5** Sata
- P4** Sata Power
- J12** Analog Input
- SW2** Upgrade (not currently implemented, reserved for future expansion)
- J9** Power Port
- CON25** LCD
- SW3** Back
- SW4** Home
- SW5** Menu
- J3** OTG

**J12**

- 1 A0
- 2 A1
- 3 A2
- 4 A4
- 5 A4
- 6 A5

**J9**

- 1 (NC)
- 2 IOREF
- 3 RESET
- 4 3.3V DC output
- 5 5V DC output
- 6 GND
- 7 GND
- 8 +5V IN



**Nota aclaratoria:** El microcomputador pcDuino posee memoria Flash de 2GB en la cual está instalada una versión de Ubuntu por defecto. Esta característica permite utilizar el pcDuino para aplicaciones sencillas, con solo conectar periféricos y el cable de alimentación a cinco voltios, que no requieran demasiado espacio de almacenamiento o exijan excesivos recursos al sistema.

Cuando las aplicaciones son exigentes en cuanto a uso de memoria, velocidad y espacio de almacenamiento, es recomendable instalar y bootear el sistema operativo desde una memoria micro SD.

En la imagen de la distribución de pines, es fácilmente reconocible la función de cada conector y cada puerto de esta versión de la tarjeta pcDuino.

Figura 27

Interfaz gráfica del sistema operativo del pcDuino  
Fuente: Elaboración propia.



**Nota aclaratoria:** Para el caso de las aplicaciones propuestas en este tutorial, es suficiente con la versión de Ubuntu preinstalada por defecto. Por lo tanto, conecte los periféricos del sistema (Hub USB, teclado, ratón, monitor y cable Ethernet) y alimente la tarjeta con 5V usando el conector OTG. Inmediatamente empezará a cargar el sistema operativo Ubuntu (distribución de Linux) y su entorno gráfico.

Para mayor información, consulte las siguientes páginas:

<http://learn.linksprite.com/pcduino/quick-start/explanation-of-pcduino3-headers/>

## 1.1 Cambiar configuración de teclado

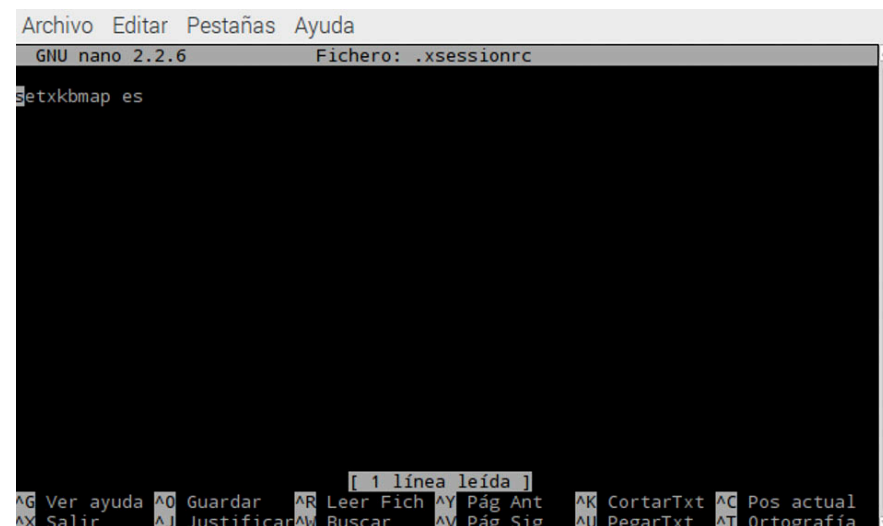


Figura 28

Terminal de configuración del sistema operativo de la pcDuino  
 Fuente: Elaboración propia.

**Nota aclaratoria:** Si aparecen símbolos extraños cuando escribe con el teclado, realice la configuración del teclado para el entorno gráfico. En sistemas Linux es frecuente hablar del "terminal" o la "consola" para referirse al entorno de comandos, el entorno 'no gráfico' o el entorno base de Linux.

**Paso 1** Para activar una ventana de terminal, siga la siguiente ruta: *Inicio*, menú *Accesorios*, opción *LXTerminal*. En adelante, los comandos requeridos en una ventana de terminal tendrán formato de letra blanca sobre fondo negro.

### Paso 2

Active una ventana de terminal.

Debe mostrar la ubicación actual dentro del sistema de carpetas, que inicialmente será el directorio de usuario, presentando el nombre del directorio: `ubuntu@ubuntu: ~ $`

### Paso 3

Cree el archivo de configuración del entorno gráfico y ábralo para editarlo con Nano, digitando:

**`sudo nano .xsessionrc`**

Dentro del archivo, escriba:

**`setxkbmap es`**

Guarde y salga: (Ctrl+o, enter, Ctrl+x)

## 2. Configuración adicional

### 2.1 Instalar utilidad de gestión de paquetes para Python

Se requiere tener acceso a Internet.

La utilidad PIP permite instalar y gestionar paquetes de instalación de *software* de manera fácil y rápida desde una ventana de terminal. Para instalarla, digite los siguientes comandos en secuencia:

**`$ sudo apt-get install python-pip python-dev build-essential`**

**`$ sudo pip install --upgrade pip`**

**`$ sudo pip install --upgrade virtualenv`**

(SaltyCrane Blog, 2010).

## 2.2 Instalar utilidad con sistema de control de versiones

Uno de los más populares sistemas de control de versiones es Git. El repositorio de archivos más extendido para este sistema es GitHub.

Para instalar *Git*, digite:

```
sudo apt-get update  
sudo apt-get install git
```

(Digital Ocean, 2014).

## 2.3 Instalar utilidades para manipular archivos .xls con Python

En una ventana de terminal, descargue las librerías e instálelas digitando:

```
sudo pip install xlwt  
Sudo pip install xlrd  
Sudo pip install xlutils
```

(*pip* es una utilidad de instalación)  
 xlwt contiene rutinas de escritura.  
 xlrd contiene rutinas de lectura.  
 xlutils contiene rutina de manipulación de archivos.

Para mayor información y ejemplos, consulte la página:

<https://ubuntulife.wordpress.com/2011/09/25/crear-leer-y-modificar-una-hoja-excel-con-python/>



## Ejemplo

```
#Código en Python que crea archivo, inserta hoja "Datos", inserta información y guarda el archivo .xls.  
import xlwt #Importa librería de escritura  
import time #Importa librería de manipulación de tiempo y fechas  
wb = xlwt.Workbook() #Crea Nuevo espacio de trabajo  
ws = wb.add_sheet('Datos',cell_overwrite_ok=True) #Inserta nueva hoja llamada "Datos" y habilita escritura  
ws.write(0, 0, 'FECHA', style0) #En la celda A1 (fila0,col0) escribe "FECHA"  
ws.write(1, 0, datetime.now(), style1) #En la celda A2 escribe la fecha actual  
ws.write(2, 0, 4)  
ws.write(2, 1, 1) #En la celda B3 escribe "1"  
ws.write(2, 2, xlwt.Formula("A3+B3"))  
wb.save('Ejemplo.xls') #Guarda el archivo con nombre "Ejemplo.xls"
```

## 2.4 Configurar para subir archivos a dropbox desde Python

### Paso 1

Ingresa a la página de Dropbox para desarrolladores (<https://www.dropbox.com/developers>) a la cuenta de usuario creada previamente

**Nombre de usuario:** Proydom@yahoo.com  
**Clave:** rasp2017

Seleccione el cuadro *Read our docs* para buscar información de la instalación en algún dispositivo de cómputo.

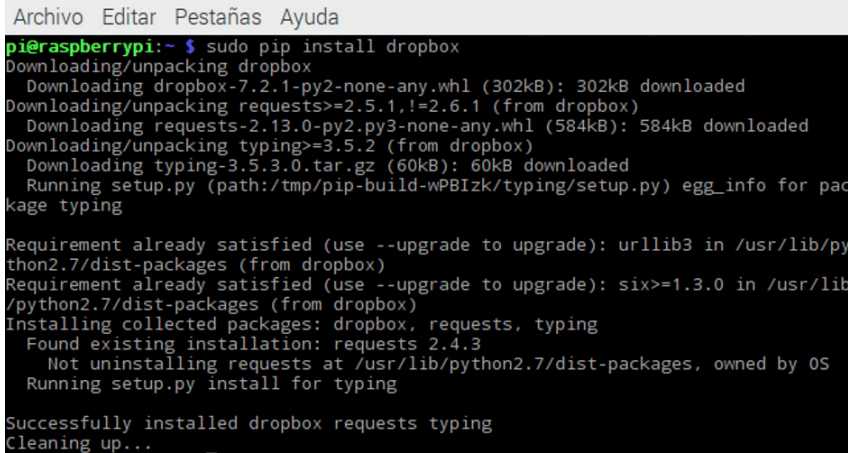
**Paso 2** Seleccione la plataforma con la cual se va a programar. En este caso Python.

**Paso 3** En la pestaña *Install*, en la sección *Install the Python SDK*, Se informa que para descargar e instalar el SDK debe utilizar el siguiente comando en una ventana de terminal:

**Sudo pip install Dropbox**

**Paso 4** En el código Python que está diseñando, importe la librería Dropbox, incluyendo esta línea en la cabecera:

```
import Dropbox.
```



**Figura 29**

Instalación del paquete de Dropbox en pcDuino.  
Fuente: Elaboración propia.

**Paso 5** Para acceder desde Python a los archivos de Dropbox en la nube sin requerir autenticación en cada proceso, use un *token de acceso* (código de acceso directo).

Para obtenerlo, es necesario que vaya nuevamente a la página de Dropbox para desarrolladores y elija el cuadro *Create your app*.

**Paso 6** En la página *Create a new app on the Dropbox Platform*, en el numeral 1, seleccione *Dropbox API*.

En el numeral 2, elija *Full Dropbox- Access to all files and folders in a user's Dropbox*.

En el numeral 3, asigne un nombre a la aplicación (no acepta nombres repetidos).

En la nueva ventana de la aplicación, en la pestaña *Settings*, busque el campo *Generated access token* y haga clic en *Generate*.

Como resultado, Dropbox proporciona un *Access Token* de ingreso exclusivo.

En este caso, para la aplicación *Sensor\_Hum\_Tem.py* generó el token:

**XE-R9TnDEEAAAAAAAAAAC6UFxNuopcz-ngitvVQauZ4A3iY20Z6Lx6kiRa0s9Pa4e**



De esta manera, cuando Python requiera acceder a Dropbox, tendrá que hacerlo mediante la estructura:

```
dbx=dropbox.Dropbox(XE-R9Tn-  
DEEAAAAAAAAAAC6UFxNuopczngi-  
tvVQauZ4A3iY20Z6Lx6kiRa0s9Pa4e)
```

# Ahora dbx tiene las propiedades:

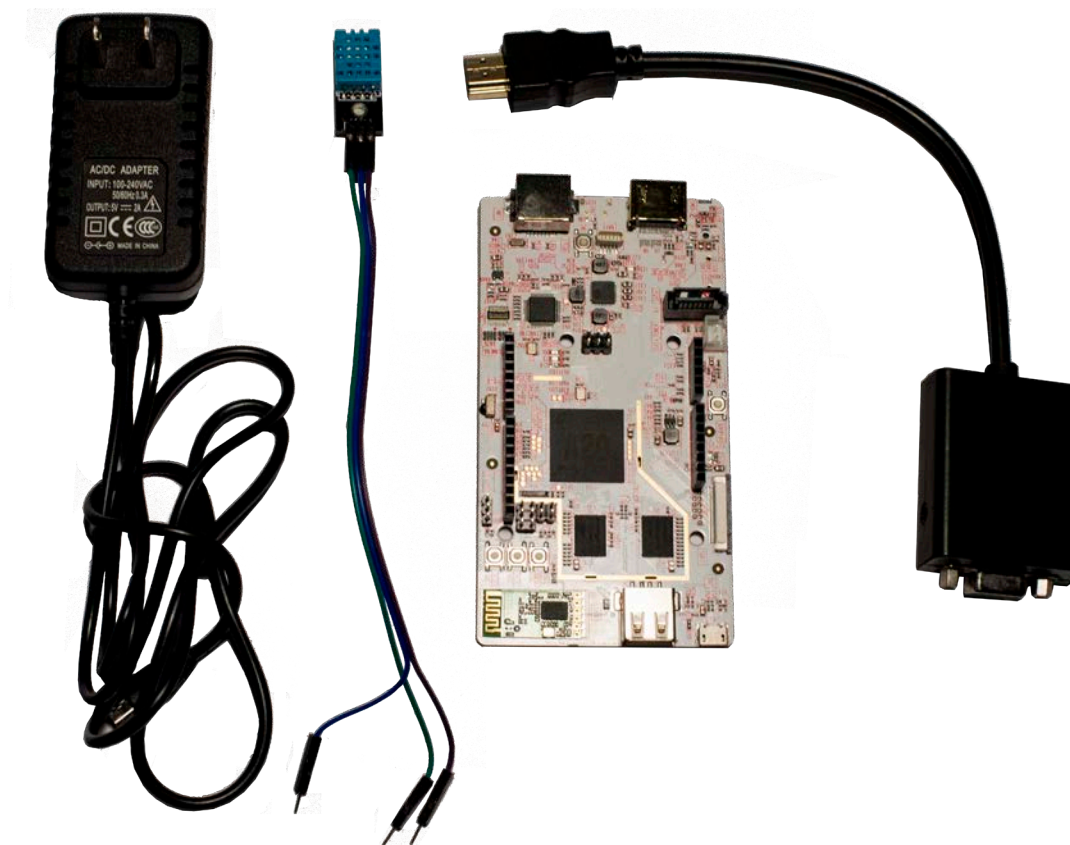
```
dbx.files_upload(EstructuraDeDatos,  
                  'ruta en Dropbox', modo, mute)  
# Para subir archivos a Dropbox
```

```
dbx.files_download_to_file('Archivo-  
Destino.xls', 'ruta en Dropbox')  
# Para descargar archivos
```

80

### 3. Aplicación: subir datos a la nube (medición de temperatura/humedad)

- **Materiales:**
  - \* Sistema microcomputador Raspberry Pi 3 (con periféricos conectados)
  - \* Sensor de temperatura y humedad relativa DHT11
  - \* Cables de conexión rápida
  - \* Fuente de poder (5V, 1A)



81

**Figura 30**  
Componentes electrónicos para el desarrollo de la aplicación.  
Fuente: Elaboración propia.

### 3.1 El sensor DHT11

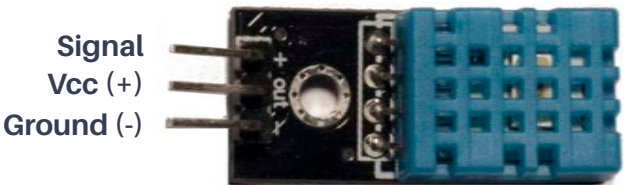
**Presentación:** Este sensor (Figura 31) sirve para medir tanto la humedad relativa como la temperatura. Tiene resolución de 1% para la humedad relativa y de 1°C para la temperatura. Se caracteriza por tener la señal digital calibrada mediante un microcontrolador de 8 bits integrado, lo que asegura alta calidad y fiabilidad en las lecturas. Está constituido por dos sensores resistivos (NTC y humedad).

Rango de medición de humedad: 20% - 95%

Rango de medición de temperatura: 0°C - 50°C

El protocolo de comunicación es serial a través de un único hilo (protocolo 1-wire) y con capacidad para transmitir la señal hasta 20 metros de distancia.

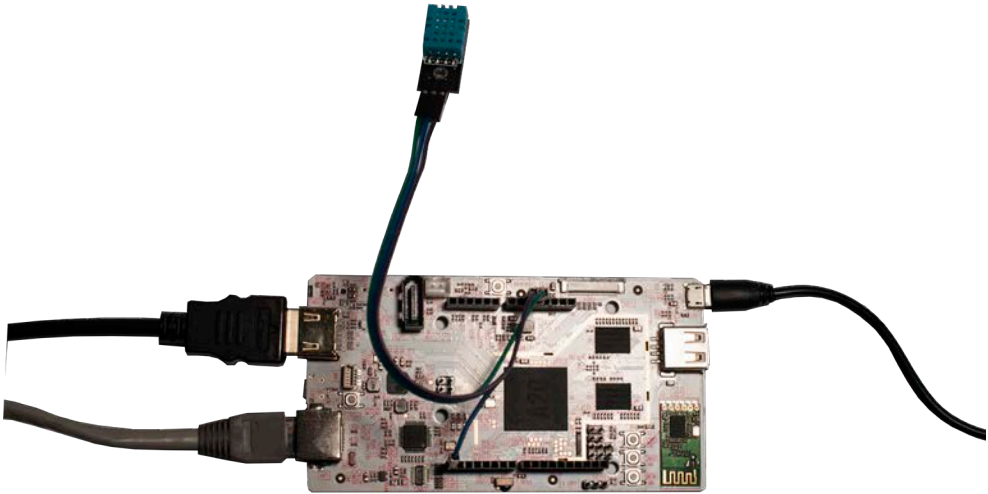
10k Ohm Pull Up Resistor



**Figura 31**

Sensor de temperatura y humedad DHT11  
Fuente: Elaboración propia.

**Paso 1** Conectamos el sensor a la pcDuino como se muestra en la Figura 32.



**Figura 32**

Montaje entre el pcDuino y el sensor de DTH11  
Fuente: Elaboración propia.

- Terminal Negativo del sensor - Conector J9 - Pin 6 de pcDuino
- Terminal Positivo del sensor - Conector J9 - Pin 5 de pcDuino
- Terminal Data del sensor - Conector J11 - Pin 1 de pcDuino (gpio 0)

**Paso 2** Instalar librería para el sensor DHT11

Se utiliza una librería de la casa Adafruit para este tipo de sensores.

En una ventana de terminal, ingrese el comando para clonar el directorio:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

Ingresa al directorio recién copiado:

```
cd Adafruit_Python_DHT
```

Instale la librería con los comandos:

```
sudo apt-get install build-essential python-dev
```

Y a continuación digite:

```
sudo python setup.py install
```

### Paso 3 Programa en Python

Para escribir el código en el entorno gráfico de *Ubuntu*, abra un archivo nuevo en el editor de texto por defecto, haga clic en el botón de inicio en la parte inferior izquierda del escritorio, menú *Accesorios*, y seleccione *Leafpad*.

Se inicia una ventana del editor para trabajar sobre un archivo nuevo (*Untitled*). Seleccione menú *File*, opción *Save* y en la ventana *Save As* asigne el nombre de archivo (*name*) y escoja un directorio para guardar (preferiblemente en directorio *ubuntu*). Guarde (opción contigua al botón *Cancel*).

En el documento, ingrese el siguiente código:

```
#Programa que lee datos de temperatura y humedad desde un sensor DHT11,  
#guarda los datos en un archivo .xls y lo sube a la nube a una carpeta de Dropbox
```

```
#!/usr/bin/env python  
import gpio  
import sys  
import Adafruit_DHT  
import xlwt  
import time  
import dropbox
```

```
MaxLect = 10#60  
Lect = 1
```

```
print "Inicio .xls"  
wb = xlwt.Workbook()  
ws = wb.add_sheet('Datos',cell_overwrite_ok=True)  
ws.write(0, 0, 'ID')  
ws.write(0, 1, 'TEMPERATURA')  
ws.write(0, 2, 'HUMEDAD')  
while(Lect <= MaxLect):  
    ws.write(Lect, 0, Lect)  
    ws.write(Lect, 1, temperatura)  
    ws.write(Lect, 2, humedad)  
    time.sleep(5)  
    Lect = Lect + 1  
wb.save('HumTem.xls')  
print "Fin .xls"
```

```
dbx = dropbox.Dropbox('XE-R9TnDEEAAAAAAAAAAAE3e_XDY2iN1uP48ZWsWJjokwfzKu92cr-  
GPvW9MEJum41')  
fi = open ('HumTem.xls')  
str = fi.read()  
dbx.files_upload(str, 'HumTem.xls', dropbox.files.WriteMode.overwrite, mute = True)  
fi.close()  
print "Archivo .xls en Dropbox"
```

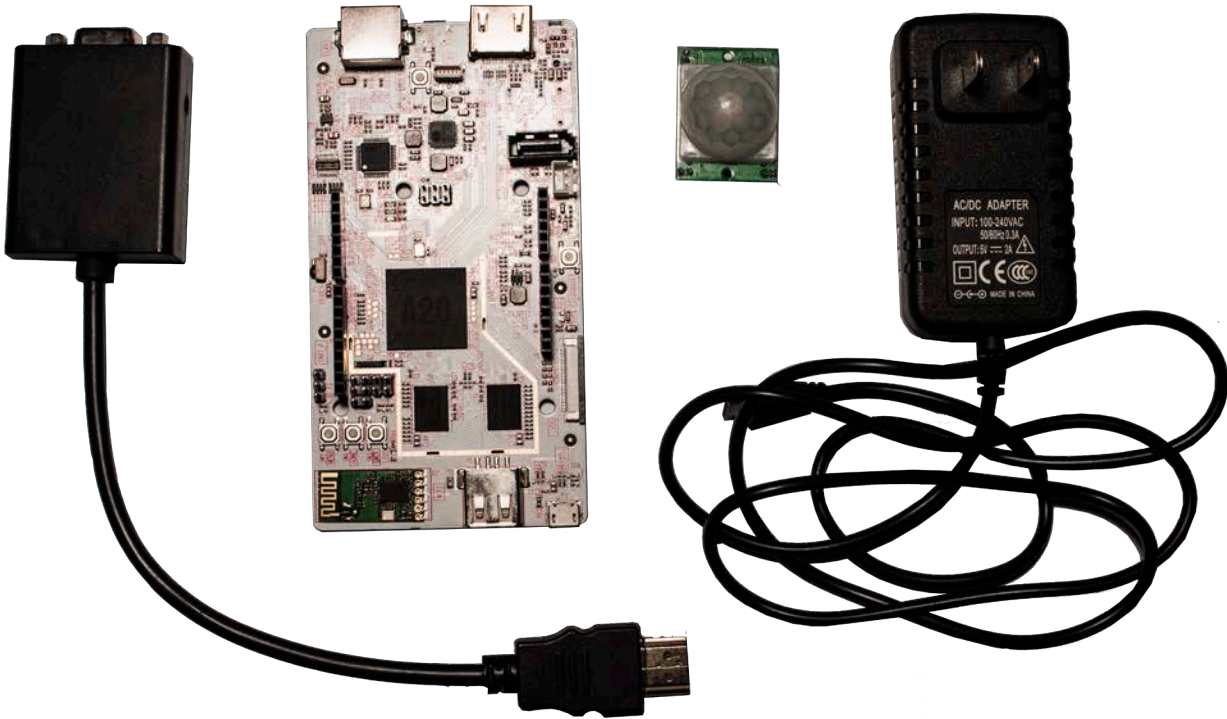
Guarde con el nombre *HumTem.py*  
En una ventana de terminal, ingrese el comando:

```
Sudo python HumTem.py
```

Verifique que en la cuenta de Dropbox se haya creado el archivo *HumTem.xls*.

4. Aplicación: subir datos a la nube  
(detección de presencia de personas)

- **Materiales:**
- \* Sistema microcomputador pcDuino3 (con pe-  
riféricos conectados)
  - \* Sensor de PIR HC-SR501
  - \* Cables de conexión rápida
  - \* Fuente de poder (5V, 1A)



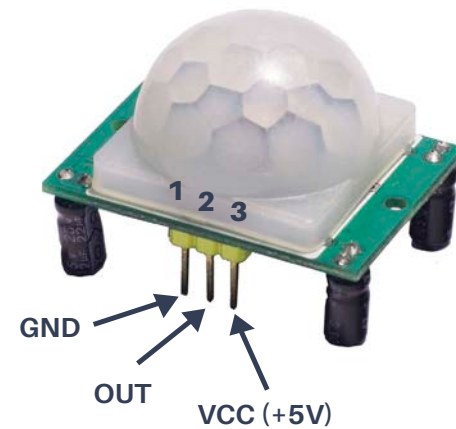
**Figura 33**  
Dispositivos de conexión del sensor de presencia y pcDuino.  
Fuente: Elaboración propia.

## El sensor HC-SR501

**Presentación** Este sensor detecta cambios en la radiación infrarroja que recibe, activando una salida de control.

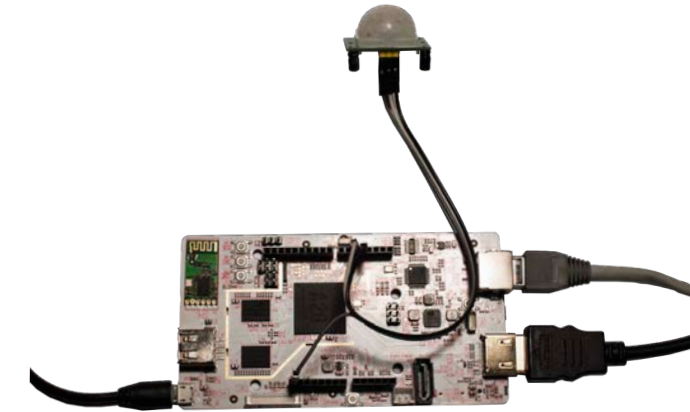
Los PIR detectan el movimiento comparando el estado de dos ventanas IR. Los objetos estáticos frente al sensor generan patrones similares en las dos ventanas, mientras que los objetos en movimiento generan patrones diferentes, cuanto más rápido se mueven. Poseen una lente Fresnell para mejorar el ángulo de detección al dividir el área de detección en sectores con diferentes lentes.

Se alimentan con 5V y la señal de salida tiene nivel de 3.3V.



**Figura 34**  
Sensor de presencia HC-SR501  
Fuente: Elaboración propia.

**Paso 1** Conecte el sensor a la pcDuino (Figura 35) siguiendo la numeración física de los pines.



**Figura 35**  
Montaje del sensor de presencia y pcDuino  
Fuente: Elaboración propia.

- Terminal Negativo del sensor - Conector J9 - Pin 6 de pcDuino
- Terminal Positivo del sensor - Conector J9 - Pin 5 de pcDuino
- Terminal Data del sensor - Conector J11 - Pin 1 de pcDuino (gpio 0)

## 4.1 Clonar librería para manipular puertos de Entrada / Salida

**Paso 2** En GitHub se encuentra un directorio con los archivos necesarios para manipular los puertos I/O digitales y análogos con código Python.

Clone el directorio con el comando:

**git clone <https://github.com/pcduino/python-pcduino>**



Verifique que se ha creado el directorio `python-pcduino` con el comando:

**ls**

Cambie al directorio que contiene el archivo de prueba `blink_led.py` para encender y apagar un led:

**cd python-pcduino/Samples/blink\_led**

Este programa enciende y apaga el led conectado a la salida GPIO18, ubicado en la tarjeta cerca del conector P7.

En este directorio se encuentra la carpeta `gpio`, cópiela en la misma ubicación del programa `.py` que desee ejecutar.



**Figura 36**

Montaje de conexión entre pcDuino y el computador de escritorio  
Fuente: Elaboración propia.

### Paso 3 Programa en Python

Para escribir el código en el entorno gráfico de *Ubuntu*, abra un archivo nuevo en el editor de texto por defecto, haga clic en el botón de inicio en la parte inferior izquierda del escritorio, menú *Accesorios*, y seleccione *Leafpad*.

Se inicia una ventana del editor para trabajar sobre un archivo nuevo (*Untitled*). Abra el menú *File*, opción *Save* y en la ventana *Save As* asigne el nombre de archivo (*name*) y escoja un directorio para guardar (preferiblemente *ubuntu*). Guarde (contiguo al botón *Cancel*).

En el documento, ingrese el siguiente código:

```
#Programa que lee datos de un sensor PIR HC-SR501, guarda los datos en un archivo .xls  
#y lo sube a la nube a una carpeta de Dropbox
```

```
#!/usr/bin/env python  
import gpio  
import sys  
import xlwt  
import time  
import dropbox
```

```
led_pin = "gpio 18"  
sensor = "gpio 0"
```

```
gpio.pinMode(led_pin, gpio.OUTPUT)  
gpio.pinMode(sensor, gpio.INPUT)  
current_state = 0
```

```
MaxLect = 10 #60
Lect = 1

print "Inicio .xls"
wb = xlwt.Workbook()
ws = wb.add_sheet('Datos',cell_overwrite_ok=True)
ws.write(0, 0, 'ID')
ws.write(0, 1, 'HORA')
ws.write(0, 2, 'MOVIMIENTO')

while(Lect <= MaxLect):
    time.sleep(0.1)
    current_state = gpio.digitalRead(sensor)
    print "Movimiento detectado"
    print "", time.strftime("%X")

    if (gpio.digitalRead(sensor) == gpio.HIGH):
        print "Movimiento detectado"
        print "", time.strftime("%X")
        #gpio.digitalWrite(led_pin, gpio.HIGH)
        ws.write(Lect, 0, Lect)
        ws.write(Lect, 1, time.strftime("%X"))
        ws.write(Lect, 2, 1)
    else:
        #gpio.digitalWrite(led_pin, gpio.LOW)
        ws.write(Lect, 0, Lect)
        ws.write(Lect, 1, time.strftime("%X"))
        ws.write(Lect, 2, 0)
```

```
time.sleep(0.1)
Lect = Lect + 1
wb.save('Movim.xls')
print "Fin .xls"
```

```
dbx = dropbox.Dropbox('XE-R9TnDEEAAAAAAAAAAAE3e_XDY2iN1uP48ZWsWJjokwfzKu92cr-
GPvW9MEJum41')
fi = open('Movim.xls')
str = fi.read()
dbx.files_upload(str, 'Movim.xls', dropbox.files.WriteMode.overwrite, mute = True)
fi.close()
print "Archivo .xls en Dropbox"
```

Guarde con el nombre *Movim.py*.

En una ventana de terminal, ingrese el comando:

**Sudo python Movim.py**

Verifique que en la cuenta de Dropbox se haya creado el archivo *Movim.xls*.

**Para mayor información, consulte las siguientes páginas:**

<https://s3.amazonaws.com/pcduino/book/Introduction+to+pcDuino.pdf>

<http://learn.linksprite.com/uncategorized/quick-start-flash-spruce-arduino-bootloader/>

<https://ubuntulife.wordpress.com/2011/09/25/crear-leer-y-modificar-una-hoja-excel-con-python/>

<https://github.com/dropbox/dropbox-sdk-python/blob/master/example/back-up-and-restore/backup-and-restore-example.py>

<http://fpaez.com/sensor-de-movimiento-infrarojo-hc-sr501/>

Usted puede acceder a cualquier sistema de almacenamiento en la nube como Dropbox o OneDrive para consultar los datos. En este caso se utilizó Dropbox.

# Conclusiones

Los sistemas embebidos son fáciles de implementar para el desarrollo de proyectos en diferentes disciplinas de la ingeniería. Específicamente, las tarjetas Raspberry Pi 3 y pcDuino, con sus componentes de *software* y *hardware*, permiten desarrollar fácilmente tareas complejas.

Durante el proceso de creación de esta cartilla, se pidió a los estudiantes que practicaran la instalación de las tarjetas para el desarrollo de proyectos, obteniéndose grandes resultados en efectividad y eficacia. Se recogieron e incorporaron sugerencias y se generó un ambiente de expectativa por su publicación.

Los autores agradecen el estímulo recibido por la comunidad académica de la Universidad Piloto de Colombia y esperan haber podido contribuir con sus procesos ingenieriles.



## Referencias bibliográficas

Blog Historia de la Informática (18 de diciembre, 2013). Raspberry Pi [mensaje de blog]. Recuperado de <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

Circuit Basics (marzo, 2016). How to Set Up the DHT11 Humidity Sensor on the Raspberry Pi [mensaje de blog]. Recuperado de <http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/>

Digital Ocean (abril, 2014). How To Install Git on Ubuntu 14.04 [mensaje de blog]. Recuperado de <https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-14-04>

Ibeyond (abril, 2016). Motion detection at your home [mensaje de blog]. Recuperado de <http://ibeyond.com/wordpress/motion-detection-home/>

Michelone, M. (marzo, 2013). pcDuino, ¿competencia de la Raspberry Pi? *Unocero*. Recuperado de <https://www.unocero.com/noticias/pcduino-competencia-de-la-raspberry-pi/>

Salas Arrián, S. (2015). *Todo sobre sistemas embebidos*. Lima: Universidad Peruana de Ciencias Aplicadas. Recuperado de <http://repositorioacademico.upc.edu.pe/upc/bitstream/10757/579143/1/Todo+sobre+sistemas+embebidos+-+1er+cap.pdf>

SaltyCrane Blog (febrero, 2010). How to Install PIP on Ubuntu [mensaje de blog]. Recuperado de <https://www.saltycrane.com/blog/2010/02/how-install-pip-ubuntu/>

Zhao, L., y. Liu, J. (s. f.). *Introduction to pcDuino*. Recuperado de <https://s3.amazonaws.com/pcduino/book/Introduction+to+pcDuino.pdf>



# Bibliografía

- Cierco, D. (Coord.) (2011). *Cloud Computing: retos y oportunidades*. Madrid: Fundación Ideas. Recuperado de [http://www.gutierrez-rubi.es/wp-content/uploads/2011/05/DT-Cloud\\_Computing-Ec.pdf](http://www.gutierrez-rubi.es/wp-content/uploads/2011/05/DT-Cloud_Computing-Ec.pdf)
- Cisco (2013). Informe anual de seguridad de Cisco 2013: La vida en el mundo actual de interconexiones. Recuperado de <https://studylib.es/download/8480071>
- Diccionario de la Real Academia Española (2015). Domótica. Recuperado de <http://dle.rae.es/?id=E7W0v9b>
- DragonBoard 410c - Qualcomm Developer Network (2010-2017). *DragonBoard 410c. A Product of Arrow Electronics*. USA: Qualcomm Developer Network. Recuperado de <https://developer.qualcomm.com/hardware/dragonboard-410c>
- Kyas, O. (2016). *How To Smart Home: A Step by Step Guide to Your Personal Internet of Things*. Wyk: Keyconcept Press. Recuperado de <http://howtosmarthome.com/>
- Lara, P. (2012). Acondicionamiento de señales [mensaje de blog]. Recuperado de <http://1538445.blogspot.com.co/2012/11/22-acondicionamiento-de-senales.html>
- Leitner, G. (2015). *The Future Home is Wise, Not Smart. A Human-Centric Perspective on Next Generation Domestic Technologies*. Klagenfurt, Austria: Springer. doi:10.1007/978-3-319-23093-1
- Lledó, E. (2012). Diseño de un sistema de control domótico basado en la plataforma Arduino (Tesis de pregrado). Escola Tècnica Superior d'Enginyeria Informàtica. Universitat Politècnica de València, Valencia, España.
- Mell, P., y Grance, T. (2011). *The NIST definition of cloud computing, recommendations of the National Institute of Standards and Technology, Special Publication 800-145*. Recuperado de <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>



- Molina, J., y Gutiérrez, A. (2016). Implementación de un prototipo de sistema domótico ahorrador de energía controlado desde aplicación móvil Android. *Revista Ingeniería y Región* 15(1), 65-74.
- Piuri, V., y Minerva, R. (julio de 2015). Construyendo la Internet de las Cosas. Computing Now [mensaje de blog]. Recuperado de <https://www.computer.org/web/computingnow/archive/building-the-internet-of-things-july-2015-spanish-version>
- Primorac, C. R. (2014). Computación en Nube (Tesis de pregrado). Universidad Nacional del Nordeste, Chaco, Argentina.
- Saito, N. (2013). Ecological Home Network : An Overview. *Proceedings of the IEEE*, 101(11), 2428-2435.
- Saito, N., y Menga, D. (2015). *Ecological Design of Smart Home Networks Technologies, Social Impact and Sustainability*. Cambridge, UK: Woodhead Publishing Series in Electronic and Optical Materials.
- Segura, A. A. (2015). Arquitectura de *Software* de Referencia para Objetos Inteligentes en Internet de las Cosas, *Revista Latinoamericana de Ingeniería de Software*, 4(2), 73-110.
- Ukil, A., Sen, J., y Koilakonda, S. (marzo, 2011). Embedded security for Internet of Things. En I. Warpakma (General Chair). 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS). IEEE, Meghalaya, India.

